

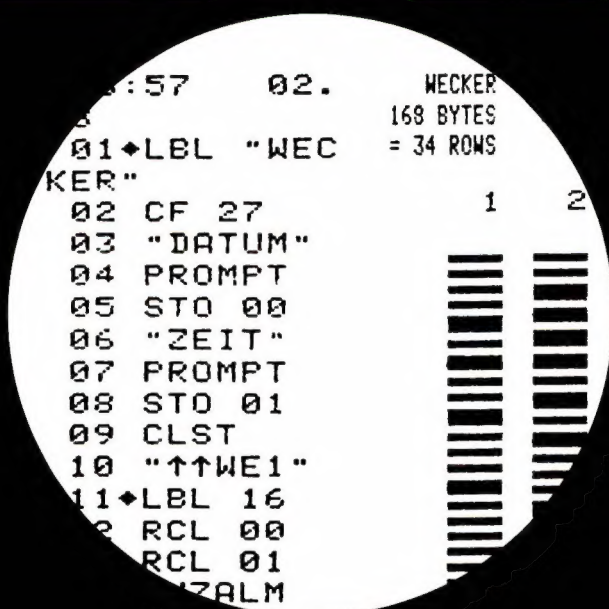
VIEWEG PROGRAMMBIBLIOTHEK

Mikrocomputer 28

HP-41

Hilfen und Anwendungen

**Speichern, Sortieren,
Mischen, Verwalten,
Rechnen**



HP-41
Hilfen und Anwendungen

Aus dem Programm Mikrocomputer

Anwenderhandbuch HP-41 C/CV

von K. Gosmann

Softwareentwicklung am Beispiel einer Dateiverwaltung

von M. Gehret

Optimales Programmieren mit dem HP-41

von G. Kruse

Aus der Vieweg Programmbibliothek Mikrocomputer zum HP-41:

Band 14

Lineares Optimieren

11 HP-41-Programme

Band 15

Dienstprogramme (Tool-Kit)

für den HP-41

Band 18

Probleme der Festigkeitslehre

23 Programme für den HP-41

Band 21

HP-41 in der Praxis

Band 23

HP-41-Sammlung

Band 27

Kryptologie-Programme

(HP-41 C/CV)

Band 28

HP-41 – Hilfen und Anwendungen

Vieweg

Vieweg Programmbibliothek
Mikrocomputer Band 28

Harald Schumny (Hrsg.)

HP-41 – Hilfen und Anwendungen

**Speichern, Sortieren,
Mischen, Verwalten,
Rechnen**

14 Programme von Peter Reiter
mit Barcodes



Friedr. Vieweg & Sohn Braunschweig / Wiesbaden

Reiter, Peter:

HP 41 — Hilfen und Anwendungen: Speichern,
Sortieren, Mischen, Verwalten, Rechnen;
14 Programme; mit Barcodes / von Peter Reiter. —
Braunschweig; Wiesbaden: Vieweg, 1985.
(Vieweg-Programmbibliothek Mikrocomputer;
Bd. 28)
ISBN 3-528-04397-0

NE: GT

Der Autor des Bandes

Peter Reiter

Herrgottwiesgasse 202
A-8055 Graz

Das in diesem Buch enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor übernimmt infolgedessen keine Verantwortung und wird keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

1985

Alle Rechte vorbehalten

© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig 1985

Die Vervielfältigung und Übertragung einzelner Textabschnitte, Zeichnungen oder Bilder, auch für Zwecke der Unterrichtsgestaltung, gestattet das Urheberrecht nur, wenn sie mit dem Verlag vorher vereinbart wurden. Im Einzelfall muß über die Zahlung einer Gebühr für die Nutzung fremden geistigen Eigentums entschieden werden. Das gilt für die Vervielfältigung durch alle Verfahren einschließlich Speicherung und jede Übertragung auf Papier, Transparente, Filme, Bänder, Platten und andere Medien. Dieser Vermerk umfaßt nicht die in den §§ 53 und 54 URG ausdrücklich erwähnten Ausnahmen.

Umschlaggestaltung: Peter Lenz, Wiesbaden

Druck und buchbinderische Verarbeitung: W. Langelüdecke, Braunschweig
Printed in Germany

ISBN 3-528-04397-0

Inhaltsverzeichnis

Vorwort	VI
Hinweise für die Programmierung	VII
1 Datenspeicherung in einem File	1
2 Sortieren von Zahlen	9
3 Fakultät bis 571 auf eine Stelle genau	13
4 Mischen von Ziffern, Zahlen oder Zeichen	20
5 Division auf unendlich	23
6 Terminverwaltung	27
7 Kuckucksuhr elektronisch	39
8 Wecken durch den HP-41CX	46
9 Löschen des erweiterten Speichers	48
10 Primfaktorzerlegung	51
11 Quersummenberechnung	54
12 Fakultäten, Variationen und Kombinationen auch für Mengen größer 69	57
13 Bruchrechnen	62
14 Brüche, schnell gekürzt	69
15 Anhang: Barcodes	72

Vorwort

In der heutigen Zeit — im Computerzeitalter — lassen sich Taschencomputer wie der HP-41CX gar nicht mehr wegdenken. Solche Kinder der Technik sind wahre Wunderwerke des Menschen und seines Denkens. Wenn man bedenkt, daß der HP-41CX vor ca. 20 Jahren so groß wie ein Einfamilienhaus gewesen wäre, kann man sich ungefähr ein Bild davon machen, welch gewaltiger Fortschritt in einem solchen elektronischen Gerät steckt.

Welchen Sinn hat aber ein „Taschenrechner“ mit annähernd 250 Operationen im täglichen Leben?

Diese und ähnliche Fragen muß man sich selbst beantworten, denn jeder Mensch hat andere Bedürfnisse und keiner sieht in einem solchen Apparat das gleiche.

Was sehen die Benutzer, Programmierer und Techniker in einem solchen Gerät?

Sicherlich nicht das Wunderding, das jene Personen in ihnen sehen, die diesen „Dingen der Zukunft“ aus dem Wege gehen, sondern ein unterstützendes und praktisches Hilfsmittel, mit dem sich außer den mathematischen Problemen noch eine Vielzahl anderer Aufgaben bewältigen lassen.

Freilich, überall sind Schranken gesetzt, so auch hier. Aber wenn man nicht an die Schranken herangeht, sie untersucht und sich mit ihnen beschäftigt, so wird man kaum jemals über sie hinwegkommen, sondern sie immer nur als Hindernis im Wege sehen.

Die Probleme die ich damit angesprochen haben möchte, lassen sich kurz erklären:

Die größte Schwierigkeit bei diesen kleinen Computern ist die Rechengeschwindigkeit. Sie läßt sich leider nicht mit dem minimalen Stromverbrauch in Einklang bringen; aber man muß ja irgendwo immer Abstriche machen. Hier möchte ich allerdings gleich erwähnen, daß der Hewlett-Packard-Rechner wohl einer der schnellsten unter den Rechnern in dieser Größen- und Leistungsklasse ist.

Irgendwann, meist nach der Zeit des Erlernens, wird sich dann das Problem des Speicherplatzes einstellen. HP hat sich schon vorher eingehend mit diesem Punkt beschäftigt und bietet neben den Speichererweiterungsmodulen auch ein Massenspeichermedium in Form eines Kassettenlaufwerks an. Zwar ist dies wohl nicht die beste Möglichkeit bezüglich Zugriffszeit, aber die Handhabung dieses Zusatzgerätes ist sehr einfach und die Bauweise äußerst robust.

Oft werden die Taschencomputer mit den diversen Mikrocomputern verglichen. Bei solchen Diskussionen hält man sich besser in seiner Meinungsäußerung zurück, außer man besitzt von beiden Klassen ein oder mehrere Exemplare. Probieren Sie es einmal aus und mischen Sie bei so einer Debatte mit; Sie werden feststellen was ich gemeint habe. Am Ende dieses Gesprächs werden Sie wissen, daß man Birnen und Zwetschgen kaum vergleichen kann.

Hinweise für die Programmierung

Leider bietet eine Schreibmaschine nicht die Möglichkeit, die gleichen Zeichen wie in der Anzeige des HP-41 darzustellen. Darum sollten Sie die folgenden Zeilen besonders gut durchlesen.

Programm-zeile	Befehl	Eingabe
XX	LBL "DIV"	(SHIFT) LBL – ALPHA – D – I – V – ALPHA
XX	APPEND " "	ALPHA – (SHIFT) K – SPACE – ALPHA (APPEND: = im Alphamodus die Zweitfunktion der Taste "K")
XX	APPEND "= "	ALPHA – (SHIFT) K – SPACE – = – ALPHA
XX	"OUTPUT"	ALPHA – O – U – T – P – U – T – ALPHA
XX	x	Das Multiplikationszeichen (*) wird im Buch durch ein kleines "x" ersetzt.

Die Ziffern, Zeichen oder Buchstaben die sich zwischen zwei Anführungszeichen befinden, sind in das ALPHA-Register einzugeben.

1 Datenspeicherung in einem File

Wie Sie sicherlich wissen werden, hat der HP-41CX max. 319 Datenspeicher. Da dies in manchen Fällen zu wenig sein kann, vor allem wenn man lange Programme abgespeichert hat, habe ich ein Programm zu Papier gebracht, mit dem Sie Daten in einem File abspeichern können, wobei auch eine Registerarithmetik möglich ist. Sollten Sie nun 2 Speichererweiterungsmodule in den Rechner stecken, so haben Sie 601 Datenregister zur Verfügung.

1.1 Programmbedienung

Bevor Sie die Speicherbefehle verwenden können, müssen Sie, sofern der File, den Sie benutzen wollen, schon vorhanden ist, den Filenamen im ALPHA-Register abspeichern und mit XEQ "FLREG" das Programm vorbereiten. Somit ist der File für die Datenspeicherung der momentane File. Sollte noch kein File für die "Registerarithmetik im erweiterten Speicher" vorhanden sein, so geben Sie im ALPHA-Register den Filenamen und im X-Stack die Anzahl der benötigten Register bekannt. Ebenso mit XEQ "FLREG" die Software starten. Nun ist der File für die Registerarithmetik vorhanden, außerdem ist dieser File der momentane File. Die einzelnen Befehle können mittels einer XEQ-Anweisung bzw. Tastenzuordnung oder über das Label "FLAR" benutzt

werden. Das Label "FLAR" ermöglicht es, in einer sich öfter wiederholenden Programmschleife verschiedene File-registeroperationen auszuführen, wobei die auszuführende Operation im ALPHA-Register bekanntgegeben wird.

1.2 Anweisungsliste

Prog.		
Zeile	Befehl	Kurzerklärung
01	LBL "FLREG"	Dieser Programmteil dient dazu, einen
02	SF 25	File anzulegen bzw., sollte dieser
03	R↑	schon vorhanden sein, diesen zum
04	CLX	momentanen File zu erklären.
05	SEEKPTA	File anlegen (noch kein File vorhanden):
06	FC?C 25	Filenamen = ALPHA-Register
07	GTO 15	Registeranzahl = X-Stack
08	RTN	File vorhanden:
09	LBL 15	Filenamen = ALPHA-Register
10	RDN	Jeweils Programmstart mit XEQ "FLREG".
11	CRFLD	
12	RTN	
13	LBL "FLAR"	Der Befehl "FLAR" ermöglicht ein
14	SF 00	indirektes Arbeiten mit den File-
15	R↑	operationen des Programms. Den zu
16	ASTO X	bearbeitenden Befehl speichern Sie im
17	GTO IND X	ALPHA-Register und starten, nachdem
		Sie die Daten je nach Operation ein-
		gegeben haben, mittels XEQ "FLAR".

Beispiel: 50,4 ist zum Inhalt des File-
registers Nr. 10 zu addieren.

37

38

39 "FL+"

40 10

41 ENTER

42 50,4

43 XEQ "FLAR"

44

45

18	LBL "FLST0"	Abspeichern von Daten im File (gleich
19	XEQ 22	dem Befehl ST0).
20	x<>y	Vor Programmstart:
21	SEEKPT	Fileregisternummer = Y-Stack
22	x<>y	Zahl = X-Stack
23	SAVEX	Nach Bearbeitung:
24	RTN	Fileregisternummer = Y-Stack
		Zahl = X-Stack
25	LBL "FLRCL"	Rückrufen von Daten aus dem File (gleich
26	XEQ 22	dem Befehl RCL).
27	SEEKPT	Vor Programmstart:
28	GETX	Fileregisternummer = X-Stack
29	RTN	Nach Bearbeitung:
		Fileregisternummer = Y-Stack
		Zahl = X-Stack
30	LBL "FL+"	Diese Befehle sind ähnlich den
31	LBL 16	Operationen ST+, ST-, STx und ST/,

32	XEQ 22	wobei hier nach der Berechnung das
33	x<>y	Ergebnis im X-Stack und im File
34	SEEKPT	enthalten ist.
35	x<>y	Vor Programmstart:
36	GETX	Fileregisternummer = Y-Stack
37	FS?C 01	Zahl = X-Stack
38	GTO 18	Nach Bearbeitung:
39	FS?C 02	Fileregisternummer = Y-Stack
40	GTO 19	Ergebnis = X-Stack
41	FS?C 03	
42	GTO 20	
43	+	
44	LBL 21	
45	x<>y	
46	SEEKPT	
47	x<>y	
48	SAVEX	
49	RTN	
50	LBL "FL-"	
51	SF 01	
52	GTO 16	
53	LBL "FLx"	
54	SF 02	
55	GTO 16	
56	LBL "FL/"	
57	SF 03	
58	GTO 16	
59	LBL 18	
60	x<>y	
61	-	
4		

62	GTO 21	
63	LBL 19	
64	x	
65	GTO 21	
66	LBL 20	
67	x<>y	
68	/	
69	GTO 21	
70	LBL 22	
71	FC?C 00	
72	RTN	
73	CLX	
74	RDN	
75	RTN	
76	LBL "FL<>RG"	(Nicht über "FLAR" möglich!)
77	XEQ 22	Dieser Programmteil tauscht einen
78	SEEKPT	Fileregisterinhalt mit dem Inhalt
79	GETX	eines Datenregisters.
80	x<> IND z	Vor Programmstart:
81	x<>y	Datenregisternummer = Y-Stack
82	SEEKPT	Fileregisternummer = X-Stack
83	x<>y	Nach Bearbeitung:
84	SAVEX	Inhalt vom Datenregister vor
85	RCL IND y	Bearbeitung = T-Stack
86	RDN	Inhalt vom Fileregister vor
87	RDN	Bearbeitung = Z-Stack
88	RTN	Datenregisternummer = Y-Stack
		Fileregisternummer = X-Stack

89	LBL "FLIND"	(Nicht über "FLAR" möglich!)
90	x<> y	Im Y-Stack geben Sie jenes File-
91	SEEKPT	register an, das die Parameter
92	x<> y	für den im ALPHA-Register spezifizierten
93	GETX	Befehl und im X-Stack gespeicherte
94	x<> y	Zahl aufweist.
95	GTO "FLAR"	Beispiel: "ST+" = ALPHA-Register

6 = Y-Stack

5 = X-Stack

Im Fileregister Nr. 6 = 4 und im
Fileregister Nr. 4 = 10 enthalten.

Im Register Nr. 6 findet der Rechner
die Parameterangabe 4, jetzt addiert
er den Registerinhalt des Fileregisters
Nr. 4 mit dem Inhalt des X-Stacks.
Das Ergebnis wird im X-Stack bzw.
Fileregister Nr. 4 abgespeichert.

Vor Programmstart:

Fileoperation = ALPHA-Register

Fileregister, das den Parameter

enthält = Y-Stack

Zahl = X-Stack

Nach Bearbeitung:

Fileregister, das den Parameter

enthalten hat = Z-Stack

Inhalt des mit dem Parameter

spezifizierten Registers = Y-Stack

Ergebnis = X-Stack

96	LBL "FL<> FL"	(Nicht über "FLAR" möglich!)
97	XEQ 22	Austauschen zweier Fileregister.
98	SEEKPT	Vor Programmstart:
99	GETX	1. Fileregisternummer = Y-Stack
100	x<> z	2. Fileregisternummer = X-Stack
101	SEEKPT	Nach Bearbeitung:
102	GETX	Inhalt 1. Fileregister vor
103	x<> z	Bearbeitung = T-Stack
104	SEEKPT	Inhalt 2. Fileregister vor
105	x<> z	Bearbeitung = Z-Stack
106	SAVEX	1. Fileregisternummer = Y-Stack
107	x<> y	2. Fileregisternummer = X-Stack
108	SEEKPT	
109	x<> t	
110	SAVEX	
111	RDN	
112	RDN	
113	RTN	
114	LBL "FLCLX"	Löschen von Registerbereichen.
115	XEQ 22	Vor Programmstart:
116	LBL 00	Das letzte zu löschende
117	SEEKPT	Register = Y-Stack
118	ENTER	das erste zu löschende
119	CLX	Register = X-Stack
120	XEQ "FLSTO"	
121	RDN	Den ganzen Inhalt des Files löschen
122	1	Sie mit dem Befehl "CLFL", der im
123	+	Benutzerhandbuch auf Seite 213

124 $x \leq y?$ beschrieben ist.

125 GTO 00

126 CLST

127 RTN

128 END Programmende

2 Sortieren von Zahlen

Diese Software verwendet als Subroutine Teile des Programms "FLREG"! Sollten Sie den Speicher mit zwei Memory-Modulen erweitert haben, so können Sie bis zu 601 Daten sortieren. Leider ist der Zeitaufwand sehr hoch.

2.1 Bedienungshinweise

Eingabe

Start mit XEQ "INPUT". Kurz darauf erscheint "HOW MANY?" in der Anzeige. Sie müssen dem Rechner nun bekanntgeben, wieviel Daten Sie sortieren wollen. Die Eingabe bestätigen Sie mit R/S. Als nächstes verlangt der Computer schon die Zahlen, wobei Sie jedesmal mit R/S abschließen.

Sortieren

Start mit XEQ "SORT", oder, wenn bei der Eingabe "END" in der Anzeige erscheint, mit R/S. Bei Beendigung des Programms ertönt ein Ton, und der Computer bleibt stehen.

Ausgabe

Die Daten rufen Sie mit "FLRCL" ab.

Beispiel: Sortieren von den folgenden 10 Zahlen:

37; 12; 8; 9; 41; 812; 7,2; 410,2; 8,1; 81.

Sortieren

EINGABE	ANZEIGE
XEQ "INPUT"	"HOW MANY?"

EINGABE	ANZEIGE
10 R/S	0,0000
37 R/S	37,0000
12 R/S	12,0000
8 R/S	8,0000
9 R/S	9,0000
41 R/S	41,0000
812 R/S	812,0000
7,2 R/S	7,2000
410,2 R/S	410,2000
8,1 R/S	8,1000
81 R/S	81,0000
XEQ "SORT"	0,0000

Nach Beendigung des Sortiervorganges

EINGABE	ANZEIGE
0 XEQ "FLRCL"	7,2000
1 XEQ "FLRCL"	8,0000
2 XEQ "FLRCL"	8,1000
3 XEQ "FLRCL"	9,0000
4 XEQ "FLRCL"	12,0000
5 XEQ "FLRCL"	37,0000
6 XEQ "FLRCL"	41,0000
7 XEQ "FLRCL"	81,0000
8 XEQ "FLRCL"	410,2000
9 XEQ "FLRCL"	812,0000

2.2 Anweisungsliste

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
1	LBL "INPUT"	27	XEQ "FLRCL"
2	"HOW MANY?"	28	x <> y
3	PROMPT	29	RDN
4	"SORT"	30	x < y?
5	SF 25	31	GTO 16
6	PRUFL	32	LBL 17
7	CF 25	33	CLA
8	XEQ "FLREG"	34	1
9	CLX	35	ST+ 01
10	LBL 00	36	RDN
11	STOP	37	RCL 01
12	SF 25	38	SF 25
13	SAVEX	39	SEEKPT
14	FS?C 25	40	RDN
15	GTO 00	41	FS?C 25
16	"END"	42	GTO 15
17	PROMPT	43	x <> y
18	LBL "SORT"	44	RCL 00
19	CLX	45	x <> y
20	STO 00	46	XEQ "FLSTO"
21	1	47	1
22	STO 01	48	ST+ 00
23	LBL 15	49	RCL 00
24	RCL 00	50	1
25	XEQ "FLRCL"	51	+
26	RCL 01	52	STO 01

Sortieren

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
53	SF 25	65	x<>y
54	SEEKPT	66	RDN
55	FS?C 25	67	x<>y
56	GTO 15	68	RCL 00
57	TONE 5	69	x<>y
58	CLST	70	XEQ "FLST0"
59	STOP	71	x<>y
60	LBL 16	72	RDN
61	x<>y	73	x<>y
62	RCL 01	74	GTO 17
63	x<>y		
64	XEQ "FLST0"	75	END

3 Fakultät bis 571 auf eine Stelle genau

3.1 Hintergrund

Stellen Sie sich vor, Sie mischen ein Kartenspiel mit 100 Karten. Was glauben Sie, wie viele Möglichkeiten gibt es, daß diese Spielkarten unterschiedlich aufeinander zu liegen kommen?

100, 1000 oder 100 000?

Es gibt wohl eine Unzahl mehr Konstellationen.

Um diese Möglichkeiten zu errechnen, können Sie nicht den Befehl "FACT" aus dem Rechner-ROM benutzen, da Ihnen der Computer mit "OUT OF RANGE" in der Anzeige stehenbleibt, außer Sie haben eine der Flags 24 bzw. 25 gesetzt. Um auch bei solchen Problemen noch ein Resultat zu erhalten und um eine höhere Genauigkeit zu erzielen, habe ich diese Software entwickelt.

Dieses Programm ermöglicht es, die Fakultät bis 571 auf eine Stelle genau zu errechnen. Es sei vor Rechenbeginn zu bedenken, daß bei Fakultät von 571 das Ergebnis nahezu 1330 Stellen aufweist. Außerdem sollten Sie sich nicht wundern, wenn Sie auf das Resultat etliche Stunden warten sollten, es geht eben nicht schneller. Sicherlich könnte man dieses Programm mit dem Worte "Spielerei" auf die Seite schieben. Dies sollte aber nicht Sinn und Zweck dieser Software sein, sondern ich wollte einfach zeigen,

wie man mit ein bißchen Kreativität die Grenzen von 10 Stellen leicht sprengen kann.

3.2 Programmbedienung und Beispiel

Die Zahl, von der die Fakultät berechnet werden soll, geben Sie in das X-Register und starten das Programm mit XEQ "xN". Wenn nun das Ergebnis ermittelt ist, stoppt der Computer und hat z. B. $630 = 4932178$ in der Anzeige stehen, d.h., daß nach der "8" noch 630 Stellen folgen. Mit R/S werden Ihnen die nächsten 7 Stellen ausgegeben, Sie müssen diesen Vorgang so oft wiederholen, bis "END" in der Anzeige erscheint. Im Normalfall weist jede Ausgabe 7 Stellen auf, außer es sind führende Nullen vorhanden, z.B. Anzeige 77 = 34206 eigentliches Ergebnis 77 = 0034206 .

Beispiel: n = 13

EINGABE	ANZEIGE
13 XEQ "xN"	7 = 622
R/S	0 = 7020800
R/S	"END"

Verwendete Register: 0 bis 4 und

10 bis 199

3.3 Anweisungsliste

Prog.		
Zeile	Befehl	Kurzbeschreibung
1	LBL "xN"	Programmbezeichnung
2	INT	Kommastellen und "User Flag" löschen
3	CF 27	
4	572	Vergleichen, ob 572 kleiner oder
5	$x <= y?$	gleich der Eingabe (mit Eingabe ist
6	GTO 22	die Zahl, aus der die Fakultät berechnet werden soll, gemeint) ist. Sollte dieser Vergleich mit ja (do if true) beantwortet werden können, so wird zu Label 22 gesprungen.
7	CLX	200 Register dem Datenspeicher
8	200	zuordnen.
9	PSIZE	
10	CLRG	Register löschen.
11	RDN	Die Eingabe in Register 0 und 10 ab-
12	STO 00	legen. Das Register 0 enthält den
13	STO 10	Multiplikationsfaktor, mit dem die Datenspeicher 10 bis 199 multipliziert werden.
14	1	1 vom Multiplikationsfaktor
15	ST- 00	(ehemalige Eingabe) subtrahieren.

16	10	10 in Register 1 speichern. Register
17	STO 01	1 dient als Datenspeicherzeiger für die indirekte Adressierung der Registerarithmetik.
18	LBL 16	Multiplikationsfaktor mit dem in
19	RCL 00	Register 1 definiertem Datenspeicher
20	STx IND 01	multiplizieren und den Überlauf
21	RCL 02	addieren. Ein Überlauf entsteht dann,
22	ST+ IND 01	wenn mehr als 7 Stellen in einem Datenspeicher (Reg. 10 - 199) sind.
23	1 E07	Den Überlauf errechnen, vom Daten-
24	ST/ IND 01	speicher subtrahieren und in Register
25	RCL IND 01	2 ablegen.
26	INT	
27	STO 02	
28	RCL IND 01	
29	FRC	
30	1 E07	
31	x	
32	STO IND 01	
33	LBL 17	Den Datenspeicher um 1 erhöhen und
34	1	abfragen ob der Zeiger auf Register
35	ST+ 01	200 weist, bei Beantwortung mit "JA"
36	RCL 01	zu Label 18 verzweigen.
37	200	
38	x = y?	
39	GTO 18	
16		

40	RCL IND 01	Vergleichen, ob der nächste Daten-
41	x≠0?	speicher ungleich 0 ist, bei "JA",
42	GTO 16	Rücksprung zu Label 16.
43	RCL 02	Überlauf im leeren Datenspeicher ab-
44	STO IND 01	legen und Überlaufregister (Reg. 2)
45	CLX	löschen.
46	STO 02	
47	1	Abfrage, ob die nächsten 10 Daten-
48	ST+ 03	speicher leer sind, ist diese Abfrage
49	RCL 03	positiv, Rücksprung zu Label 17.
50	10	
51	x≠y?	
52	GTO 17	
53	LBL 18	Register 3 löschen.
54	CLX	
55	STO 03	
56	10	Datenspeicherzeiger auf 10
57	STO 01	zurücksetzen.
58	1	Multiplikationsfaktor um 1 verringern
59	ST- 00	und vergleichen, ob er ungleich 1 ist,
60	RCL 00	bei "JA", Rücksprung zu Label 16.
61	1	
62	x≠y?	
63	GTO 16	

64	RCLFLAG	Flags 0 bis 43 im Register 4 speichern
65	STO 04	(Hier beginnt der Programmteil für die Ausgabe des Ergebnisses.)
66	FIX 0	Keine Kommastellen.
67	CF 29	Zifferntrennzeichenflag löschen.
68	199	Datenspeicherzeiger auf 199 setzen.
69	STO 01	
70	LBL 23	Datenspeicher mit 0 vergleichen, bei
71	RCL IND 01	"JA", Sprung zu Label 21. Wenn erster
72	x=0?	besetzter Datenspeicher gefunden ist,
73	GTO 21	wird eine Tonkombination abgegeben.
74	BEEP	
75	LBL 24	Außer den angezeigten Stellen die
76	CLA	noch vorhandene Stellenanzahl berechnen
77	RCL 01	und ins Alpharegister bringen.
78	10	
79	-	
80	7	
81	x	
82	ARCL X	
83	APPEND " ="	An das Alpharegister "SPACE" und "=" anhängen.

84	ARCL IND 01	Inhalt des Datenspeichers an das
85	PROMPT	Alpharegister anhängen.
86	1	Datenspeicherzeiger um 1 verringern
87	ST- 01	und vergleichen, ob ungleich 9,
88	RCL 01	bei "JA", Rücksprung zu Label 24,
89	9	ansonst Flag 0 bis 43 mit dem in
90	x≠y?	Register 4 enthaltenen Wert
91	GTO 24	restaurieren, Stack löschen, "END"
92	RCL 04	anzeigen und Programm stoppen.
93	STOFLAG	
94	CLST	
95	"END"	
96	PROMPT	
97	LBL 21	Datenspeicherzeiger um 1 verringern
98	1	und Rücksprung zu Label 23.
99	ST- 01	
100	GTO 23	
101	LBL 22	"ERROR" anzeigen und Programm stoppen.
102	"ERROR"	
103	PROMPT	
104	END	Programmende

4 Mischen von Ziffern, Zahlen oder Zeichen

4.1 Grundsätzliches

Nehmen Sie vier Blätter Papier und schreiben Sie je einen Buchstaben des Wortes "ROSE" auf ein Blatt. Jetzt vertauschen Sie die Blätter untereinander. Diese Buchstaben bilden nun maximal 24 (Fakultät aus 4 = 24) neue Wörter, wobei die meisten keine Bedeutung haben. Das gleiche läßt sich zum Beispiel mit Zahlen wiederholen.

Mit diesem Programm wird es Ihnen möglich, max. 10 Ziffern, Zahlen und Zeichen ohne großen Aufwand zu mischen. Leider ist der Zeitaufwand relativ hoch.

Beispiel: "ROSE" = ALPHA-Register
XEQ "MIS"

ROSE - ORSE - RSOE - SROE - OSRE - SORE - ROES - ORES -
REOS - EROS - OERS - EORS - RSEO - SREO - RESO - ERSO -
SERO - ESRO - OSER - SOER - OESR - EOSR - SEOR - ESOR -

Bei jeder neuen Stellung der Zeichen wird ein Ton ausgegeben und die neue Zeicheninformation in die Anzeige gebracht. Die letzte Ausgabe wird durch die BEEP-Anweisung akustisch bekanntgegeben und bleibt daraufhin noch ca. 2 Sekunden sichtbar.

4.2 Anweisungsliste

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
1	LBL "MIS"	27	RCL 14
2	CLX	28	STx 13
3	x<>F	29	RDN
4	CF 08	30	1
5	CF 09	31	-
6	,015	32	ENTER ↓
7	CLRGX	33	CLX
8	10	34	x≠y?
9	STO 14	35	GTO 01
10	ATOX	36	CLST
11	LBL 00	37	LBL 02
12	STO IND 10	38	CLX
13	1	39	1
14	ST+ 11	40	+
15	ST+ 10	41	ST+ 12
16	CLST	42	RCL 14
17	ATOX	43	STx 12
18	x≠y?	44	RDN
19	GTO 00	45	RCL 11
20	1	46	x≠y?
21	ST- 11	47	GTO 02
22	RCL 11	48	RCL 14
23	ENTER ↓	49	ST/ 12
24	LBL 01	50	LBL 18
25	RDN	51	RCL 11
26	ST+ 13	52	1

Mischen

Prog.	
Zeile	Befehl

53	+
54	STO 10
55	RCL 13
56	STO 15
57	LBL 15
58	RCL 15
59	RCL 14
60	/
61	ENTER ↴
62	INT
63	STO 15
64	RDN
65	FRC
66	RCL 14
67	x
68	RCL 11
69	x < y?
70	GTO 17
71	RDN
72	FS? IND X
73	GTO 17
74	SF IND X
75	RCL IND X
76	XTOA
77	DSE 10
78	GTO 15
79	AVIEW

22

Prog.	
Zeile	Befehl

80	TONE 5
81	CLA
82	CLX
83	x <> F
84	CF 08
85	CF 09
86	LBL 17
87	CLA
88	CLX
89	x <> F
90	CF 08
91	CF 09
92	1
93	ST- 13
94	RCL 13
95	RCL 12
96	x < =y?
97	GTO 18
98	BEEP
99	CLST
100	CLX
101	PSE
102	PSE
103	CLD
104	STOP
105	END

verwendete Register: 0 bis 15

5 Division auf unendlich

Um herauszufinden, ob jede Division ein periodischer Dezimalbruch ist, hat ein Kollege in Zusammenarbeit mit mir dieses Programm entwickelt.

5.1 Programmbedienung

Nachdem der Dividend im Y-Register und der Divisor im X-Register ist, starten Sie das Programm mit XEQ "DIV". Zuerst wird der ganzzahlige Teil der Division ausgegeben, danach folgen die Kommastellen. Die Ergebnisse sind jeweils 15 Sekunden lang sichtbar. Ihre Länge umfaßt außer der ersten Ausgabe ausnahmslos 10 Stellen.

Um das Programm stoppen zu können, müssen Sie warten bis der Indikator für die Flag 00 aufscheint, sobald Sie diese in der Anzeige sehen, betätigen Sie die Taste für den Dezimalpunkt.

Beispiel:	4001 ENTER	AUSGABE
	319	12
	XEQ "DIV"	5423197492
		1630094043
		8871473354
		2319749216

5.2 Anweisungsliste

Prog.		
Zeile	Befehl	Kurzbeschreibung
1	LBL "DIV"	Programmname
2	STO 02	Divisor im Register 2 und
3	RDN	Dividenden im Register 1 speichern.
4	STO 01	Register 3 und ALPHA-Register löschen.
5	CLX	
6	STO 03	
7	CLA	
8	RCLFLAG	Den Status der Flags 0 bis 43 im
9	STO 00	Register 0 speichern.
10	FIX 0	Keine Dezimalstellen anzeigen.
11	RCL 01	Division von Dividend und Divisor.
12	RCL 02	
13	/	
14	INT	Vom Ergebnis die Dezimalstellen löschen,
15	CF 29	Zifferntrennzeichenflag zurücksetzen
16	TONE 1	und einen tiefen Ton abgehen.
17	ARCL X	Den Wert im X-Stack in das ALPHA-
18	GTO 16	Register übertragen und zu Label 16 verzweigen.

19	LBL 15	Den Restwert der Division von
20	RCL 01	Dividend und Divisor errechnen, mit
21	RCL 02	10 multiplizieren und als neuen
22	MOD	Dividenden im Register 0 abspeichern.
23	10	
24	x	
25	STO 01	
26	RCL 02	Den neuen Dividenden mit dem Divisor,
27	/	dieser bleibt immer gleich, dividieren,
28	INT	die Kommastellen löschen und das
29	ARCL X	Ergebnis in das ALPHA-Register über-
		tragen.
30	1	Sollten noch keine 10 Stellen er-
31	ST+ 03	rechnet sein, zurückspringen zu
32	RCL 03	Label 15.
33	10	
34	x≠y?	
35	GTO 15	
36	CLX	Register 3 löschen und einen hohen
37	STO 03	Ton abgeben.
38	TONE 8	
39	LBL 16	Das ALPHA-Register in das Anzeige-
40	AVIEW	Register übertragen und löschen.
41	CLA	

Division

42	SF 00	Abfrage, ob die Dezimalpunktaste
43	GETKEY	gedrückt wird, bei nein, Rücksprung
44	CF 00	zu Label 15.
45	83	
46	$x \neq y?$	
47	GTO 15	
48	RCL 00	Restaurieren der Flags 0 bis 43 und
49	STOFLAG	löschen des Anzeige-, des ALPHA- und
50	CLD	der Stack-Register.
51	CLA	
52	CLST	
53	STOP	Programmstop
54	END	Programmende

6 Terminverwaltung

Durch die Gelegenheit, mittels genauer Zeit- und Datumsangaben ein Programm zu einer bestimmten Zeit zu starten, wird es möglich den Rechner als Terminkalender zu verwenden.

6.1 Die Software

Das Programm verwendet die Register 0 bis 6 und einen File im erweiterten Speicher. Die Software läßt sich in zwei große Teilprogramme unterteilen:

In "TERMIN" über diesem Label wird der Termin eingegeben und in "TERMIN?" mit diesem werden die Termine wieder abgerufen.

"TERMIN"

Wenn Sie das Programm mit XEQ "TERMIN" gestartet haben und noch kein File mit dem Namen "TERM1" im erweiterten Speicher vorhanden ist, erscheint kurz darauf "REGISTER?" in der Anzeige. Sie müssen nun bestimmen, wie groß der File für die Terminverwaltung sein soll. Die Eingabe beenden Sie mit R/S. Sollten Sie dem File mehr Register zuordnen wollen, als Speicherplatz im erweiterten Speicher frei ist, so erscheint "EXT MEM CLR" in der Anzeige. Sie

können die Anzeige mit "CLD" löschen. Nun steht im X-Stack jene Anzahl von Registern, die im erweiterten Speicher noch frei sind. Sie können, wenn Sie den vollen Speicherbereich zuteilen wollen, wieder R/S drücken, wobei "REGISTER?" nochmals in der Anzeige erscheint. Nun betätigen Sie abermals R/S und geben somit die Größe des Files bekannt. Als nächste Abfrage erscheint "TERM. DATE?" im Display. Nachdem Sie das Termindatum eingetastet haben, schließen Sie die Eingabe mit R/S ab, daraufhin erscheint "TERM. TIME?". Gleich wie beim Termindatum tätigen Sie die Eingabe der Terminzeit R/S. Sollte die Terminzeit später als 23 Uhr sein, so wird eine Sekunde lang "TOO LATE" angezeigt und zur Terminzeitabfrage zurückgesprungen. Mittels "TEXT" in der Anzeige erinnert der Computer, daß Sie auch eine Mitteilung bzw. Notiz abspeichern können. Geben Sie nun den Text ein (Achtung max. 24 Zeichen!) und schließen Sie mit R/S ab. Es erscheint die Frage "END J/N?". Betätigen Sie "N" für "NEIN", können Sie weiteren Text abspeichern. Diesen Vorgang können Sie so oft wiederholen, bis die Maximallänge eines Records erreicht ist. Bei "JA" (Drücken auf die Taste "J") stoppt der Rechner. Sollten Sie einen weiteren Termin zum abspeichern haben, so müssen Sie das Programm erneut mittels XEQ "TERMIN" starten.

Im erweiterten Speicher werden die Termine nicht geordnet, sondern nach ihrer Eingabe abgespeichert. Jeden Tag um 23 Uhr 50 Minuten schaltet sich der Rechner ein und löscht sämtliche an diesem Tag aufgerufene Termine.

"TERMIN?"

Das Programm starten Sie mit XEQ "TERMIN?". Kurz darauf erscheint in der Anzeige "ALL? J/N", d.h., wollen Sie alle Termine oder nur jene, welche Sie mit einem Datum spezifizieren, abrufen? Sollten Sie die Taste "J" (ja) betätigen, so werden die Termine, so wie sie im File abgespeichert wurden, aufgerufen. Da im ALPHA-Register nur 24 Zeichen Platz haben, müssen Sie, nachdem Sie die 24 Zeichen gelesen haben, mit R/S abschließen. In der Anzeige erscheint darauf "LOOK/NEXT/END", wobei Sie, um weiteren Text zu bekommen, die Taste "L" für (LOOK) drücken müssen bzw. die Taste "N" für "NEXT", wenn Sie den nächsten Termin haben wollen. Um einen Programmstop zu bewirken, brauchen Sie nur die Taste "E" für "END" betätigen. Sollten Sie nun beim Beginn (Abfrage "ALL? J/N") nichts drücken, "N" drücken oder eine andere Taste betätigen, so erscheint in der Anzeige "DATUM?". Ihre Datumseingabe beenden Sie mit R/S. Wird für diesen Tag ein Termin gefunden, so erscheint die Alarmzeit mit "XX,XX WHO J/N" in der Anzeige. D.h. wollen Sie wissen, wer, wen oder was Sie zu diesem Zeitpunkt besuchen müssen, so drücken Sie "J" für "JA" und "N" für "NEIN". Wenn Sie "N" betätigen, so wird der nächste Termin gesucht, sollte wiederum einer gefunden werden, wird er gleich angezeigt, wie der vorhergehende. Sie haben von neuem die Wahlmöglichkeit, sollte keiner gefunden werden stoppt der Rechner. Bei Betätigen der "J"-Taste erscheinen die ersten 24 Zeichen vom Record, da die Anzeige nur 24 Zeichen aufnimmt. Nachdem Sie den Text

gelesen haben, schließen Sie mit R/S ab. Nun erscheint die Abfrage "LOOK/NEXT/CLR". Sollten Sie mehr Text benötigen, drücken Sie "L" (LOOK). Wenn Sie den nächsten Termin sehen wollen, sofern einer vorhanden ist, drücken Sie die Taste "N" für "NEXT"; wenn der angezeigte Termin hinfällig ist, Sie ihn somit löschen wollen, betätigen Sie "C" für "CLEAR". Der Termin wird gelöscht, und sofern noch ein Termin für diesen Tag vorhanden ist, wird dieser angezeigt, wobei Sie gleich vorgehen wie bei den vorherigen Terminen. Sollten Sie keine der drei Tasten "L", "N" oder "C" innerhalb von 10 Sekunden betätigen oder eine falsche, so erscheint wieder "LOOK/NEXT/CLR".

Terminreminderung

Eine halbe Stunde vor jedem Termin meldet sich der Rechner mit 5 BEEP-Anweisungen und fragt, ob er den Termin vergessen (Taste "N"), oder ob er ihn mit dem Text anzeigen soll (Taste "J"). Sollte in den nächsten 10 Sekunden nach dem Alarm keine Taste betätigt werden, so meldet sich der Rechner nochmals 15 Minuten vor dem Termin. Bei der zweiten und daher letzten Terminreminderung meldet sich der Computer mit 10 BEEP-Anweisungen und fragt, ob irgendeine Taste gedrückt wird, womit Sie ihm befehlen, den Termin auszugeben. Wird keine Taste betätigt, so wird der Termin automatisch gelöscht.

6.2 Anweisungsliste

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
1	LBL "TERMIN"	27	RDN
2	RCLFLAG	28	STO 01
3	STO 02	29	CLA
4	CLST	30	ARCL X
5	SF 25	31	APPEND " "
6	"TERM1"	32	APPCHR
7	SEEKPTA	33	FIX 6
8	FC?C 25	34	CLST
9	XEQ 15	35	RCL 00
10	FIX 6	36	RCL 01
11	"TERM. DATE?"	37	,3
12	PROMPT	38	HMS-
13	STO 00	39	"↑↑ TERM2"
14	CLA	40	XYZALM
15	ARCL X	41	LBL 17
16	APPEND " "	42	"TEXT"
17	APPREC	43	AON
18	FIX 2	44	STOP
19	LBL 47	45	AOFF
20	CLX	46	APPCHR
21	"TERM. TIME?"	47	LBL 18
22	PROMPT	48	"END J/N?"
23	RND	49	AVIEW
24	23	50	GETKEY
25	x < y?	51	CLD
26	GTO 48	52	41

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
53	x=y?	80	LBL 16
54	GTO 17	81	"EXT MEM CLR"
55	RDN	82	PROMPT
56	25	83	GTO 15
57	x≠y?	84	LBL "TERM2"
58	GTO 18	85	TIME
59	CLD	86	STO 01
60	RCL 02	87	RCLFLAG
61	STOFLAG	88	STO 02
62	CLA	89	5
63	CLST	90	STO 00
64	STOP	91	LBL 20
65	LBL 15	92	BEEP
66	"REGISTER?"	93	DSE 00
67	PROMPT	94	GTO 20
68	EMROOM	95	"OUTP. J/N?"
69	x < y?	96	AVIEW
70	GTO 16	97	GETKEY
71	RDN	98	25
72	"TERM1"	99	x=y?
73	CRFLAS	100	GTO 21
74	CLST	101	RDN
75	DATE	102	41
76	23,5	103	x=y?
77	"↑↑ TERM4"	104	OFF
78	XYZALM	105	CLD
79	RTN	106	CLST

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
107	DATE	134	OFF
108	RCL 01	135	FIX 2
109	FIX 2	136	RCL 01
110	RND	137	RND
111	,15	138	,15
112	HMS+	139	HMS+
113	"↑↑ TERM3"	140	GTO 22
114	XYZALM	141	LBL 21
115	RCL 02	142	FIX 2
116	STOFLAG	143	RCL 01
117	CLST	144	RND
118	OFF	145	,3
119	LBL "TERM3"	146	HMS+
120	TIME	147	LBL 22
121	STO 01	148	CLD
122	RCLFLAG	149	STO 01
123	STO 02	150	"TERM1"
124	10	151	CLST
125	STO 00	152	SEEKPTA
126	LBL 25	153	CLA
127	BEEP	154	FIX 6
128	DSE 00	155	DATE
129	GTO 25	156	ARCL X
130	"TERMIN"	157	APPEND " "
131	AVIEW	158	FIX 2
132	GETKEY	159	RCL 01
133	x=0?	160	ARCL X

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
161	POSFL	188	RCL 00
162	RCL 02	189	ARCL X
163	STOFLAG	190	LBL 24
164	CLST	191	POSFL
165	RCLPT	192	-1
166	INT	193	x=y?
167	LBL 25	194	GTO 26
168	SF 25	195	RCLPT
169	GETREC	196	FRC
170	RCLPT	197	x≠0?
171	INT	198	GTO 24
172	x≠y?	199	DELREC
173	GTO 50	200	GTO 24
174	FC?C 25	201	LBL 26
175	GTO 50	202	CLST
176	PROMPT	203	RCL 00
177	GTO 23	204	1
178	LBL "TERM4"	205	DATE+
179	DATE	206	23,5
180	STO 00	207	" TERM4"
181	RCLFLAG	208	XYZALM
182	STO 02	209	RCL 02
183	"TERM1"	210	STOFLAG
184	CLST	211	CLST
185	SEEKPTA	212	CLA
186	FIX 6	213	OFF
187	CLA	214	LBL "TERMIN?"

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
215	"ALL? J/N"	242	RCL 00
216	AVIEW	243	x=y?
217	GETKEY	244	GTO 30
218	CLD	245	LBL 33
219	25	246	1
220	x=y?	247	ST+ 01
221	GTO 39	248	GTO 31
222	RCLFLAG	249	LBL 30
223	STO 05	250	ATOX
224	"DATUM?"	251	ATOX
225	PROMPT	252	ASTO X
226	STO 00	253	RCL 02
227	1	254	x=y?
228	STO 01	255	GTO 38
229	"TERM2"	256	RDN
230	ASTO 02	257	RCL 03
231	"TERM3"	258	x≠y?
232	ASTO 03	259	GTO 33
233	LBL 31	260	,15
234	FIX 6	261	STO 06
235	RCL 01	262	LBL 36
236	SF 25	263	RCL 04
237	RCLALM	264	x <> y
238	FC?C 25	265	HMS+
239	GTO 35	266	STO 04
240	STO 04	267	LBL 32
241	RDN	268	CLA

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
269	FIX 2	296	RCL 06
270	ARCL 04	297	HMS+
271	APPEND " WHO J/N"	298	ARCL X
272	AVIEW	299	POSFL
273	GETKEY	300	RCLPT
274	CLD	301	INT
275	25	302	STO 04
276	x=y?	303	LBL 37
277	GTO 34	304	SF 25
278	RDN	305	GETREC
279	41	306	RCL 04
280	x≠y?	307	RCLPT
281	GTO 32	308	INT
282	GTO 33	309	x≠y?
283	LBL 34	310	GTO 49
284	FIX 6	311	FC?C 25
285	"TERM1"	312	GTO 49
286	CLST	313	PROMPT
287	SEEKPTA	314	LBL 46
288	RCL 01	315	"LOOK/NEXT/CLR"
289	RCLALM	316	AVIEW
290	CLA	317	GETKEY
291	x<>y	318	CLD
292	ARCL X	319	33
293	APPEND " "	320	x=y?
294	x<>y	321	GTO 37
295	FIX 2	322	RDN

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
323	41	350	PROMPT
324	$x=y?$	351	"LOOK/NEXT/END"
325	GTO 33	352	AVIEW
326	RDN	353	GETKEY
327	13	354	CLD
328	$x\neq y?$	355	CLA
329	GTO 46	356	33
330	RCL 01	357	$x=y?$
331	CLALMX	358	GTO 40
332	GTO 31	359	RDN
333	LBL 35	360	41
334	RCL 05	361	$x=y?$
335	STOFLAG	362	GTO 41
336	CLST	363	RDN
337	CLA	364	15
338	STOP	365	$x\neq y?$
339	LBL 38	366	GTO 39
340	,3	367	CLST
341	STO 06	368	STOP
342	GTO 36	369	LBL 41
343	LBL 39	370	1
344	"TERM1"	371	ST+ 02
345	CLST	372	RCL 02
346	SEEKPTA	373	SEEKPT
347	STO 02	374	GTO 40
348	LBL 40	375	LBL 48
349	GETREC	376	"TOO LATE"

Terminverwaltung

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
377	AVIEW	386	GTO 33
378	PSE	387	LBL 50
379	CLD	388	RCL 02
380	GTO 47	389	STOFLAG
381	LBL 49	390	CLST
382	"TEXTEND"	391	CLA
383	AVIEW	392	STOP
384	PSE	393	END
385	CLD		

7 Kuckucksuhr elektronisch

Der Vorteil des Rechners gegenüber der Kuckucksuhr liegt darin, daß man den "Kuckucksruf" bei Bedarf abstellen kann. Außerdem besteht die Möglichkeit, bei jeder vollen, jeder halben oder bei jeder viertel Stunde eine Tonkombination ertönen zu lassen.

7.1 Programmbedienung

Start mit XEQ "ZEITSIG".

Folgende Fragen müssen Sie dem Computer beantworten:

"ERSTES SIG" Eingabe der Uhrzeit, wann der Rechner
 das erste Signal abgeben soll. (R/S)

"LETZTES SIG" Eingabe jener Uhrzeit, bei der das letzte
 Signal abgegeben werden soll.

"1/4H? J/N" Alle viertel Stunden ein Signal?
 "J" für ja und "N" für nein betätigen!

oder

"1/2H? J/N" Alle halben Stunden ein Signal?
 "J" für ja bzw. "N" für nein drücken.

"RTN/OFF?" Nach dem Zeitsignal: soll der Rechner
 den Befehl "RTN" oder "OFF" ausführen?
 Jeweils den Anfangsbuchstaben für die
 gewünscht Anweisung betätigen.

Verwendete Flags: 0 = Zeitintervall 1/4h
1 = Zeitintervall 1/2h
2 = Ton für die 1/2h

Verwendete Flags: 3 = Ton für die 1/4h
4 = RTN / OFF

Verwendete Register: 00 = erstes Zeitsignal
01 = letztes Zeitsignal
02 = Zeitintervall
03 = Uhrzeit für Ton
04 = Flags vom Zeitsignalprogramm
05 = Flags vom Anwender
06 = ersten 6 Alphazeichen
07 = zweiten 6 Alphazeichen
08 = dritten 6 Alphazeichen
09 = vierten 6 Alphazeichen
10 = Inhalt vom T-Stack
11 = Inhalt vom Z-Stack
12 = Inhalt vom Y-Stack
13 = Inhalt vom X-Stack
14 = Inhalt vom Last X Register
15 = Zwischenspeicher (Arbeitsreg.)

LBL "ZE1" = für Stunden

LBL "ZE2" = für halbe Stunden

LBL "ZE3" = für viertel Stunden

7.2 Anweisungsliste

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
1	LBL "ZEITSIG"	26	LBL 00
2	CF 27	27	"1/4H? J/N"
3	SF 25	28	AVIEW
4	"↑↑ ZE1"	29	GETKEY
5	CLALMA	30	25
6	SF 25	31	x=y?
7	"↑↑ ZE2"	32	GTO 16
8	CLALMA	33	RDN
9	SF 25	34	41
10	"↑↑ ZE3"	35	x≠y?
11	CLALMA	36	GTO 00
12	CF 25	37	LBL 01
13	CLST	38	"1/2H? J/N"
14	x<>F	39	AVIEW
15	,014	40	GETKEY
16	CLRGX	41	25
17	"ERSTES SIG"	42	x=y?
18	PROMPT	43	GTO 17
19	INT	44	RDN
20	STO 00	45	41
21	STO 03	46	x≠y?
22	"LETZTES SIG"	47	GTO 01
23	PROMPT	48	LBL 18
24	INT	49	"RTN/OFF?"
25	STO 01	50	AVIEW

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
51	GETKEY	78	$x > y?$
52	42	79	GTO 02
53	$x=y?$	80	RCL 03
54	GTO 36	81	XEQ 25
55	RDN	82	XEQ 26
56	52	83	LBL 30
57	$x \neq y?$	84	CLST
58	GTO 18	85	$x < > F$
59	LBL 37	86	STO 04
60	CLD	87	CLST
61	FS? 00	88	CLA
62	GTO 20	89	STOP
63	FS? 01	90	LBL 36
64	GTO 21	91	SF 04
65	1	92	GTO 37
66	LBL 22	93	LBL 43
67	STO 02	94	XEQ 32
68	TIME	95	OFF
69	RCL 01	96	LBL 16
70	$x < y?$	97	SF 00
71	GTO 19	98	GTO 18
72	LBL 02	99	LBL 17
73	RCL 03	100	SF 01
74	RCL 02	101	GTO 18
75	HMS+	102	LBL 20
76	STO 03	103	,15
77	TIME	104	GTO 22

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
105	LBL 21	132	RCL 03
106	,3	133	RCL 02
107	GTO 22	134	HMS+
108	LBL 19	135	STO 03
109	30	136	RCL 01
110	GTO 45	137	x < y?
111	LBL 41	138	GTO 41
112	42	139	RCL 03
113	LBL 45	140	XEQ 25
114	STO 15	141	XEQ 26
115	CLST	142	LBL 42
116	DATE	143	FS? 04
117	1	144	GTO 43
118	DATE+	145	XEQ 32
119	RCL 00	146	RTN
120	STO 03	147	LBL "ZE2"
121	"↑↑ ZE1"	148	TONE 8
122	XYZALM	149	TONE 4
123	GTO IND 15	150	TONE 8
124	LBL "ZE1"	151	GTO 40
125	TONE 5	152	LBL "ZE3"
126	TONE 6	153	TONE 7
127	TONE 7	154	TONE 3
128	TONE 8	155	GTO 40
129	TONE 3	156	LBL 32
130	LBL 40	157	CLA
131	XEQ 31	158	ARCL 06

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
159	ARCL 07	186	ASTO 07
160	ARCL 08	187	ASHF
161	ARCL 09	188	ASTO 08
162	CLST	189	ASHF
163	x<> F	190	ASTO 09
164	STO 04	191	CLA
165	RCL 05	192	RCLFLAG
166	STOFLAG	193	STO 05
167	RCL 14	194	RCL 04
168	STO L	195	x<> F
169	RCL 10	196	CLST
170	RCL 11	197	RTN
171	RCL 12	198	LBL 25
172	RCL 13	199	FRC
173	RTN	200	,3
174	LBL 31	201	x=y?
175	STO 13	202	GTO 23
176	RCL L	203	RDN
177	STO 14	204	,15
178	RDN	205	x=y?
179	STO 12	206	GTO 24
180	RDN	207	RDN
181	STO 11	208	,45
182	RDN	209	x=y?
183	STO 10	210	GTO 24
184	ASTO 06	211	CF 02
185	ASHF	212	CF 03

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
213	CLST	227	GTO 28
214	RTN	228	"↑↑ ZE1"
215	LBL 23	229	LBL 29
216	SF 02	230	CLST
217	CF 03	231	RCL 03
218	RTN	232	XYZALM
219	LBL 24	233	RTN
220	SF 03	234	LBL 27
221	CF 02	235	"↑↑ ZE2"
222	RTN	236	GTO 29
223	LBL 26	237	LBL 28
224	FS? 02	238	"↑↑ ZE3"
225	GTO 27	239	GTO 29
226	FS? 03	240	END

8 Wecken durch den HP-41 CX

Eine der vielen Möglichkeiten, sich aus dem Schlummer reißen zu lassen, ist jene, den Wiederholungsintervall des Alarms immer kürzer zu setzen. Genau das wird von diesem Programm gemacht und zwar im Abstand von 4, 3, 2 und 1 Minute.

8.1 Programmbedienung

Start mit XEQ "WECKER". Die Eingabe vom "DATUM" und "ZEIT" mit R/S abschließen.

Um den Wecker abzustellen, betätigen Sie beim "Läuten" die R/S- oder die ON/OFF-Taste, und Sie werden keinen Ton von Ihrem Rechner mehr hören. Sollten Sie hingegen sich nicht um die Piepserei kümmern und keine Taste drücken, so meldet sich der Computer bis zu viermal wieder.

Verwendete Register: 0 bis 2

8.2 Anweisungsliste

Prog. Zeile	Befehl	Prog. Zeile	Befehl
1	LBL "WECKER"	31	x
2	CF 27	32	INT
3	"DATUM"	33	100
4	PROMPT	34	/
5	STO 00	35	RCL 02
6	"ZEIT"	36	HMS+
7	PROMPT	37	STO 01
8	STO 01	38	CLST
9	CLST	39	XEQ 16
10	"WE1"	40	OFF
11	LBL 16	41	LBL "WE2"
12	RCL 00	42	,03
13	RCL 01	43	"WE3"
14	XYZALM	44	GTO 99
15	CLST	45	LBL "WE3"
16	CLA	46	,02
17	RTN	47	"WE4"
18	LBL "WE1"	48	GTO 99
19	,04	49	LBL "WE4"
20	"WE2"	50	,01
21	LBL 99	51	"WE5"
22	STO 02	52	GTO 99
23	15	53	LBL "WE5"
24	LBL 00	54	15
25	TONE 8	55	LBL 01
26	TONE 1	56	BEEP
27	DSE X	57	DSE X
28	GTO 00	58	GTO 01
29	TIME	59	OFF
30	100	60	END

9 Löschen des erweiterten Speichers

Um einen File aus dem erweiterten Speicher zu entfernen, mußte man ihn bisher im ALPHA-Register definieren und mit dem Befehl "PURFL" löschen. Dies kann, wenn man sehr viele Files hat, zu einer zeitraubenden Tätigkeit werden.

Die beiden Programme "CLEMX" (clear extended memory by X) und "CLEM" (clear extended memory) helfen Ihnen, diesen Aufwand zu minimieren.

"CLEM" löscht den gesamten erweiterten Speicher, und bei "CLEMX" geben Sie mit einer Zahl im X-Stack bekannt, wie groß der Speicherplatz ist, den Sie benötigen.

9.1 Programmbeschreibung

Die beiden Programme sind als Subroutinen ausgelegt, d.h. sollten Sie "CLEM" oder "CLEMX" mit einer XEQ-Anweisung anspringen, so wird das Programm ausgeführt, bei Ende desselben zum nächstenfolgenden Befehl nach XEQ "?" zurückgesprungen und das Hauptprogramm weiterbearbeitet. Sollten Sie hingegen im Rechenmodus die Anweisung XEQ "CLEM" bzw. XEQ "CLEMX" geben, so wird der RTN-Befehl zum Schluß als "STOP" interpretiert.

Bei "CLEMX" steht nach der Beendigung des Programms im X-Stack, wieviele Register Sie nun wirklich zur Verfügung haben, da das Programm von vorne nach hinten, somit in der

Reihenfolge wie bei dem Befehl "CAT 4" ausgegeben, die Files aus dem erweiterten Speicher löscht und vergleicht, ob der erforderliche Platz schon frei ist.

9.2 Anweisungsliste

Prog.		
Zeile	Befehl	Kurzbeschreibung
1	LBL "CLEM"	Programmname
2	CLST	Stack löschen.
3	1	Erstes File vom externen Speicher in
4	EMDIRX	das X- und ALPHA-Register übertragen.
5	x=y?	Wenn kein File mehr vorhanden ist,
6	RTN	zurückspringen zu vorher benützter Unterprogramm- oder Programmebene bzw. Programmstop
7	CLX	File aus dem erweiterten Speicher
8	PURFL	löschen. Rücksprung zu Programm-
9	GTO "CLEM"	anfang.
10	END	Programmende

Löschen

Prog.		
Zeile	Befehl	Kurzbeschreibung
1	LBL "CLEMX"	Programmname
2	EMROOM	Ist der Paum im erweiterten Speicher
3	x > y?	größer oder gleich der Eingabe? Wenn
4	RTN	ja, Rücksprung zu vorheriger
5	x=y?	Unterprogramm- oder Programmebene
6	RTN	bzw. Programmstop.
7	RDN	Erstes File vom erweiterten Speicher
8	1	in das X- und ALPHA-Register über-
9	EMDIRX	tragen.
10	CLX	File aus dem erweiterten Speicher
11	SF 25	löschen. Sollte kein File mehr im
12	PURFL	erweiterten Speicher sein, Rücksprung
13	RDN	zu vorheriger Unterprogramm- oder
14	EMROOM	Programmebene bzw. Programmstop.
15	FC?C 25	
16	RTN	
17	RDN	Rücksprung zu Programmanfang.
18	GTO "CLEMX"	
19	END	Programmende

10 Primfaktorzerlegung

Dieses Programm zerlegt jede Zahl in ihre Primzahlen.

10.1 Programmbedienung

Die zu zerlegende Zahl in das X-Register eingeben und das Programm mit XEQ "PRIM" starten.

Jeder Primfaktor außer dem letzten wird 1 Sekunde lang angezeigt.

Verwendete Register: 16 bis 19

Beispiel: 39240 XEQ "PRIM"

AUSGABE
2
2
2
3
5
109

10.2 Anweisungsliste

Prog.		
Zeile	Befehl	Kurzbeschreibung
1	LBL PRIM"	Programmname
2	CF 27	USERS-Flag löschen und Eingabe in
3	STO 16	Register 16 speichern.
4	LBL Ø3	Von der Zahl im X-Stack die Quadrat-
5	SQRT	wurzel ziehen, die Kommastellen
6	INT	löschen und das Ergebnis im
7	STO 19	Register 19 speichern.
8	FS? ØØ	Wurde gerade ein Primfaktor ausgegeben?
9	GTO Ø2	Wenn ja, Sprung zu Label Ø2, bei nein
10	1	1 in Register und 2 in Register 17
11	STO 18	speichern. (Reg. 17 = Divisionsfaktor)
12	2	
13	STO 17	
14	LBL Ø2	Flag Ø löschen.
15	CF ØØ	
16	LBL ØØ	Ist der Divisionsfaktor größer als
17	RCL 16	die Wurzel aus der Eingabe? Wenn ja,
18	RCL 17	Sprung zu Label 15, ansonst Division
19	RCL 19	von Eingabe mit Divisionsfaktor.

20	x < y?	Ganzzahligen Teil, des Ergebnisses
21	GTO 15	löschen. Sind Kommastellen vorhanden?
22	RDN	Wenn ja, Sprung zu Label 01, sonst
23	/	Divisionsfaktor um 1 erhöhen und zu
24	FRC	Label 00 zurückspringen.
25	x=0?	
26	GTO 01	
27	RCL 18	
28	ST+ 17	
29	GTO 00	
30	LBL 01	Primfaktor 1 sec. lang anzeigen,
31	RCL 17	Eingabe mit Primfaktor dividieren,
32	PSE	Ergebnis in den X-Stack bringen
33	RCL 17	und im Register 16 speichern. Flag 00
34	ST/16	setzen und Rücksprung zu Label 03.
35	SF 00	
36	RCL 16	
37	GTO 03	
38	LBL 15	Flag 00 löschen, letzten Primfaktor
39	CF 00	ins X-Register übertragen und
40	RCL 16	Programmstop.
41	RTN	
42	END	Programmende

11 Quersummenberechnung

11.1 Grundsätzliches

Für ein paar mathematische Tests habe ich dieses Programm zu Papier gebracht.

Wenn Sie eine beliebige Zahl, z.B. 179, notieren, so haben Sie die Zahlen 1, 7 und 9 zu Papier gebracht. Diese Zahlen addiert der Rechner. Bei unserem Beispiel hätten wir als Ergebnis 17 erhalten. Die Maximallänge ist die Stellenzahl der Mantisse. Sollte eine Zahl Kommastellen aufweisen, so wird das Komma so lange nach rechts verschoben, bis kein Dezimalbruch mehr vorhanden ist.

Dateneingabe

Diejenige Zahl, von der Sie die Quersumme errechnet haben wollen, geben Sie in den X-Stack und starten das Programm mit XEQ "QSUM". Sobald der Computer das Ergebnis gefunden hat, bleibt er mit dem Ergebnis in der Anzeige stehen. Das Erwähnenswerte bei dieser Software ist wohl, daß keine Datenspeicher zur Datenzwischenspeicherung verwendet werden, sondern nur mit den Stack-Registern gearbeitet wird.

11.2 Anweisungsliste

Prog.		
Zeile	Befehl	Kurzbeschreibung
1	LBL "QSUM"	Programmname
2	LBL 01	Sollte eine Zahl Kommastellen aufweisen, so wird diese so lange mit 10 multipliziert, bis kein Dezimalbruch mehr vorhanden ist.
3	ENTER	
4	FRC	
5	x=0?	
6	GTO 00	
7	CLX	
8	10	
9	x	
10	GTO 01	
11	LBL 00	Löschen des X- und Z-Stacks und ein Verschieben der Stack-Inhalte nach unten.
12	CLX	
13	STO z	
14	RDN	
15	LBL 03	Eingabe mit 10 dividieren und im Z-Stack speichern.
16	10	
17	/	
18	STO z	Den ganzzahligen Anteil der im X-Stack enthaltenen Zahl löschen, den Dezimalbruch mit 10 multiplizieren und den Y-Stack mit der Zahl addieren.
19	FRC	
20	10	
21	x	
22	ST+ y	

23	CLX	X-Stack löschen. (Mit diesem Befehl wird auch der Stack-Lift gesperrt.)
24	RCL z	Den Z-Stackinhalt in den X-Stack über-
25	INT	tragen, die Kommastellen löschen und
26	x≠0?	vergleichen, ob die im X-Stack
27	GTO 03	enthaltene Zahl ungleich 0 ist. Bei
28	RCL y	positiver Beantwortung der Abfrage,
29	RTN	Rücksprung zu Label 03, sonst Ausgabe der Quersumme.
30	END	Programmende

12 Fakultäten, Variationen und Kombinationen auch für Mengen größer 69

12.1 Grundsätzliches, Beispiele

Berechnung der Anzahl der möglichen Kombinationen und Variationen von n Objekten, wenn r Objekte gleichzeitig entnommen werden. Dazu Berechnung der Fakultät von positiven ganzen Zahlen.

Dieses Programm führt drei Funktionen aus.

Fakultät: Berechnung der Fakultät ($n!$) einer positiven ganzen Zahl.

$$n! = (n) (n - 1) (n - 2) \dots (n - (n-1))$$

Variationen: Berechnung der Anzahl der möglichen Variationen von n Objekten, wenn r Objekte gleichzeitig entnommen werden.

$$P = n! / (n - r)!$$

Kombinationen: Berechnung der Anzahl der möglichen Kombinationen von n Objekten, wenn r Objekte gleichzeitig entnommen werden.

$$C = n! / r! (n - r)!$$

Beispiel für Fakultät:

Die einzelnen Bücher eines zwanzigbändigen Lexikons werden in zufälliger Reihenfolge im Regal aneinandergestellt. Wieviele mögliche Anordnungen gibt es?

EINGABE	ANZEIGE
Start mit	
XEQ "FAK"	N?
20 R/S	Anzeige der Uhrzeit und nach einem Piepston das Ergebnis, in unserem Fall 2,432902 18

Es gibt $2,432902 \cdot 10^{18}$ mögliche Anordnungen.

Beispiel für Variationen:

37 Studenten legen eine Prüfung für die Bewilligung eines Stipendiums ab. Die Studenten mit den 4 besten Prüfungsergebnissen erhalten Stipendien von S 400000, S 300000, S 200000 bzw. S 100000. Wieviele unterschiedliche Resultate sind möglich? (S = Schilling)

EINGABE	ANZEIGE
Start mit	
XEQ "VAR"	N?
37 R/S	R?
4 R/S	Anzeige der Uhrzeit und nach einem Piepston das Ergebnis, in unserem Fall 1585080

Beispiel für Kombinationen:

Beim Kartengeben erhält ein Spieler 4 Karten aus einem
110-iger Blatt. Wieviele mögliche Hände kann er bekommen?

EINGABE	ANZEIGE
Start mit	
XEQ "KOM"	N?
110 R/S	R?
4 R/S	Anzeige der Uhrzeit und nach einem Piepston das Ergebnis, in unserem Fall 5773187

Sollte zum Beispiel die Fakultät von $n=70$ errechnet werden,
so wird das Ergebnis wie folgend ausgegeben:

MAN= 1,197857 R/S drücken

EXP= 100

$n! = 1,197857 \cdot 10^{100}$

12.2 Anweisungsliste

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
1	LBL "FAK"	7	GTO 18
2	CF 25	8	LBL "VAR"
3	XEQ 16	9	CF 27
4	XEQ 21	10	XEQ 16
5	RCL 00	11	XEQ 17
6	XEQ 15	12	XEQ 21

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
13	XEQ 19	40	LBL 17
14	RCL 00	41	"R?"
15	XEQ 15	42	PROMPT
16	RCL 04	43	ABS
17	-	44	INT
18	GTO 18	45	STO 01
19	LBL "KOM"	46	RCL 00
20	CF 27	47	x > y?
21	XEQ 16	48	RTN
22	XEQ 17	49	"N > R???"
23	XEQ 21	50	PROMPT
24	XEQ 19	51	LBL 15
25	RCL 01	52	STO 02
26	XEQ 15	53	CLX
27	ST+ 04	54	STO 03
28	RCL 00	55	LBL 00
29	XEQ 15	56	RCL 02
30	RCL 04	57	LOG
31	-	58	ST+ 03
32	GTO 18	59	DSE 02
33	LBL 16	60	GTO 00
34	"N?"	61	RCL 03
35	PROMPT	62	RTN
36	ABS	63	LBL 18
37	INT	64	TONE 5
38	STO 00	65	STO 02
39	RTN	66	INT
60			

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
67	SF 25	87	RCL 02
68	10 ↑ X	88	INT
69	FC?C 25	89	"EXP= "
70	GTO 01	90	ARCL X
71	RCL 02	91	AVIEW
72	FRC	92	RTN
73	10 ↑ X	93	LBL 19
74	x	94	RCL 00
75	,5	95	RCL 01
76	+	96	-
77	INT	97	XEQ 15
78	CLD	98	STO 04
79	STOP	99	RTN
80	LBL 01	100	LBL 21
81	RCL 02	101	CLA
82	FRC	102	TIME
83	10 ↑ X	103	ATIME
84	"MAN= "	104	AVIEW
85	ARCL X	105	RTN
86	PROMPT	106	END

13 Bruchrechnen

13.1 Vorbemerkungen, Beispiele

Es wird wohl ein jeder mit der Achsel zucken, wenn Sie ihn fragen, wie das Ergebnis von $7 \frac{5}{10}$ lautet.

Sie wissen natürlich in Kürze, sofern Sie richtig getippt haben, daß das Resultat nur $\frac{50}{7}$ bzw. $7 \frac{1}{7}$ sein kann.

Eine jede Dezimalzahl läßt sich als Bruch darstellen, man muß nur wissen wie.

z.B.: $50,14 = 50 \frac{14}{100}$ oder

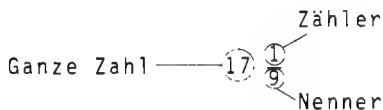
$$\frac{5014}{100} \text{ oder}$$

$$\frac{2507}{50}$$

$$17 = \frac{17}{1}$$

$$3,333 = 3 \frac{333}{1000}$$

Wie heißen die Zahlen bei einem Bruch?



Wie können Sie nun mit diesem Programm Bruchrechnen?

Anhand von Beispielen sei dies erklärt !

$$14 \frac{3}{7} : \frac{1}{6} =$$

EINGABE	ANZEIGE
Start mit	
XEQ "BRE1"	
(BRE ist die	
Abkürzung für	
<u>Bruchrechnen</u>)	GANZE ZAHL=
14 R/S	ZAEHLER=
3 R/S	NENNER=
7 R/S	GANZE ZAHL=
R/S	ZAEHLER=
1 R/S	NENNER=
6 R/S	BEFEHL?
XEQ "B/"	ZAEHLER= 606
R/S	NENNER= 7
R/S	GANZE ZAHL= 86
R/S	ZAEHLER= 4
R/S	NENNER= 7

$$\text{Ergebnis} = \frac{606}{7} = 86 \frac{4}{7}$$

Neues Beispiel:

$$1 \frac{7}{69} \cdot 13 \frac{7}{9} - \frac{18}{111} =$$

(Achtung: Punkt- vor Strichrechnung)

EINGABE	ANZEIGE
XEQ "BRE1"	GANZE ZAHL=
1 R/S	ZAehler=
7 R/S	NENNER=
69 R/S	GANZE ZAHL=
13 R/S	ZAehler=
7 R/S	NENNER=
9 R/S	BEFEHL?
XEQ "Bx"	ZAehler= 9424
XEQ "BRE2"	GANZE ZAHL=
R/S	ZAehler=
18 R/S	NENNER=
111 R/S	BEFEHL?
XEQ "B-"	ZAehler= 344962
R/S	NENNER= 22977
R/S	GANZE ZAHL= 15
R/S	ZAehler= 307
R/S	NENNER= 22977

$$\text{Ergebnis} = \frac{344962}{22977} = 15 \frac{307}{22977}$$

ALPHA-Labels und ihre Bedeutung

LBL "BRE1" = Programmstart

LBL "BRE2" = dient zur Eingabe, wenn mehrere Rechenvorgänge
bei einer Rechnung erforderlich sind.

LBL "B+" = addieren	}	der jeweiligen Brüche
LBL "B-" = subtrahieren		
LBL "Bx" = multiplizieren		
LBL "B/" = dividieren		

13.2 Anweisungsliste

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
1	LBL "BRE1"	26	ABS
2	CLST	27	RCL 02
3	XEQ 15	28	x
4	PROMPT	29	ST+ 01
5	STO 00	30	FS?C 00
6	XEQ 16	31	XEQ 05
7	PROMPT	32	RCL 03
8	STO 01	33	x < 0?
9	XEQ 17	34	SF 00
10	PROMPT	35	ABS
11	STO 02	36	RCL 05
12	LBL "BRE2"	37	x
13	CLST	38	ST+ 04
14	XEQ 15	39	FS?C 00
15	PROMPT	40	XEQ 06
16	STO 03	41	CLST
17	XEQ 16	42	STO 00
18	PROMPT	43	STO 03
19	STO 04	44	"BEFEHL?"
20	XEQ 17	45	PROMPT
21	PROMPT	46	LBL 05
22	STO 05	47	-1
23	RCL 00	48	STx 01
24	x < 0?	49	RTN
25	SF 00	50	LBL 06

Bruchrechnen

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
51	-1	78	STx 02
52	STx 04	79	LBL 19
53	RTN	80	CF 00
54	LBL "B+"	81	RCL 01
55	XEQ 18	82	x < 0?
56	RCL 01	83	XEQ 20
57	RCL 04	84	STO 03
58	+	85	RCL 02
59	STO 01	86	STO 04
60	GTO 19	87	LBL 00
61	LBL "B-"	88	CLST
62	XEQ 18	89	RCL 02
63	RCL 01	90	x=y?
64	RCL 04	91	GTO 01
65	-	92	RCL 01
66	STO 01	93	RCL 02
67	GTO 19	94	/
68	LBL "Bx"	95	FRC
69	RCL 04	96	RCL 02
70	STx 01	97	x
71	RCL 05	98	,1
72	STx 02	99	+
73	GTO 19	100	INT
74	LBL "B/"	101	x <> 02
75	RCL 05	102	STO 01
76	STx 01	103	GTO 00
77	RCL 04	104	LBL 01

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
105	RCL 01	132	STO 00
106	STO 02	133	FRC
107	RCL 03	134	RCL 02
108	x <> y	135	x
109	/	136	RCLFLAG
110	STO 01	137	x <> y
111	RCL 04	138	FIX 0
112	RCL 02	139	RND
113	/	140	x <> y
114	STO 02	141	STOFLAG
115	LBL 02	142	x <> y
116	RCL 01	143	STO 01
117	LBL 03	144	RCL 00
118	FS?C 00	145	INT
119	CHS	146	STO 00
120	STO 01	147	LBL 04
121	XEQ 16	148	RCL 00
122	XEQ 25	149	XEQ 15
123	PROMPT	150	XEQ 25
124	RCL 02	151	PROMPT
125	XEQ 17	152	RCL 01
126	XEQ 25	153	ABS
127	AVIEW	154	XEQ 03
128	RTN	155	STOP
129	RCL 01	156	GTO 04
130	RCL 02	157	LBL 18
131	/	158	RCL 05

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
159	RCL 02	176	ABS
160	STx 04	177	STO 01
161	STx 05	178	SF 00
162	RDN	179	RTN
163	STx 01	180	LBL 25
164	STx 02	181	RCLFLAG
165	RTN	182	x<>y
166	LBL 15	183	FIX 0
167	"GANZE ZAHL= "	184	CF 29
168	RTN	185	ARCL X
169	LBL 16	186	x<>y
170	"ZAEHLER= "	187	STOFLAG
171	RTN	188	x<>y
172	LBL 17	189	RTN
173	"NENNER= "	190	END
174	RTN		
175	LBL 20		

14 Brüche, schnell gekürzt

Brüche kürzen ist nicht jedermanns Sache. Um diese lästige Tätigkeit so angenehm wie möglich zu machen, sollten Sie dieses Programm benutzen.

14.1 Programmbedienung

Erklärung an Hand eines Beispieles:

Der Bruch $\frac{40318}{714}$ ist zu kürzen.

EINGABE	ANZEIGE
Start mit	
XEQ "BRUCH"	ZAEHLER=
40318 R/S	NENNER=
714 R/S	GGT= 2,0000
R/S	ZAEHLER= 20159,0000
R/S	NENNER= 357,0000

Der größte gemeinsame Teiler von 40318 und 714 ist 2, somit ist das Ergebnis $\frac{20159}{357}$.

verwendete Register : 1 bis 4

14.2 Anweisungsliste

Prog.		Prog.	
Zeile	Befehl	Zeile	Befehl
1	LBL "BRUCH"	26	GTO 00
2	XEQ 02	27	LBL 01
3	PROMPT	28	RCL 03
4	STO 01	29	RCL 01
5	STO 03	30	/
6	XEQ 03	31	STO 03
7	PROMPT	32	RCL 04
8	STO 02	33	RCL 01
9	STO 04	34	/
10	LBL 00	35	STO 04
11	CLST	36	RCL 01
12	RCL 02	37	"GGT= "
13	x = y?	38	ARCL X
14	GTO 01	39	PROMPT
15	RCL 01	40	LBL 04
16	RCL 02	41	RCL 03
17	/	42	XEQ 02
18	FRC	43	ARCL X
19	RCL 02	44	PROMPT
20	x	45	RCL 04
21	,1	46	XEQ 03
22	+	47	ARCL X
23	INT	48	PROMPT
24	x <> 02	49	GTO 04
25	STO 01	50	LBL 02

Prog.	
Zeile	Befehl
51	"ZAEHLER= "
52	RTN
53	LBL Ø3
54	"NENNER= "
55	RTN
56	END












Anhang

1 Datenspeicherung in einem File

FLPEG
313 BYTES
= 63 ROWS

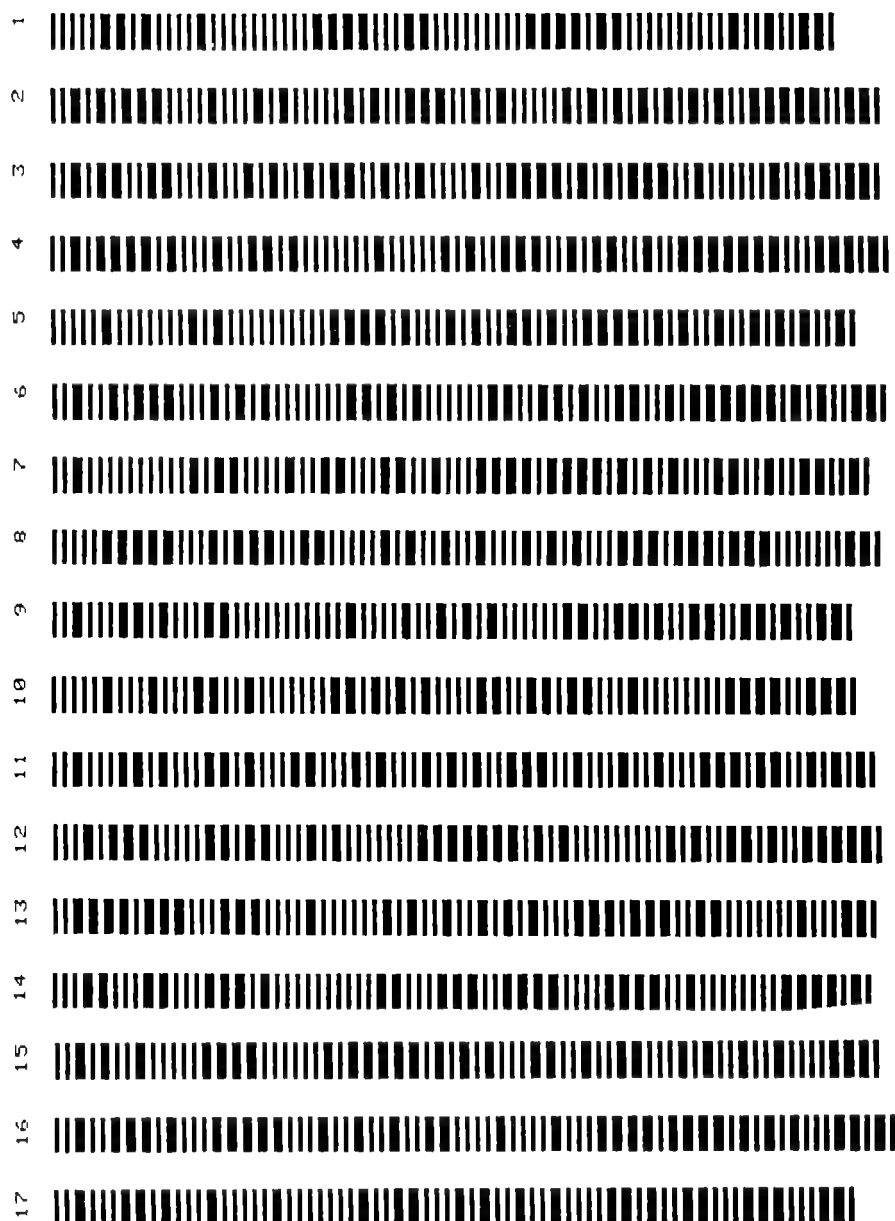
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	



















34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	

53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	

2 Sortieren von Zahlen

INPUT
171 BYTES
= 35 ROWS



18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	

3 Fakultät bis 571 auf eine Stelle genau
















*N
186 BYTES
= 38 ROWS

19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	

4 Mischen von Ziffern, Zahlen oder Zeichen

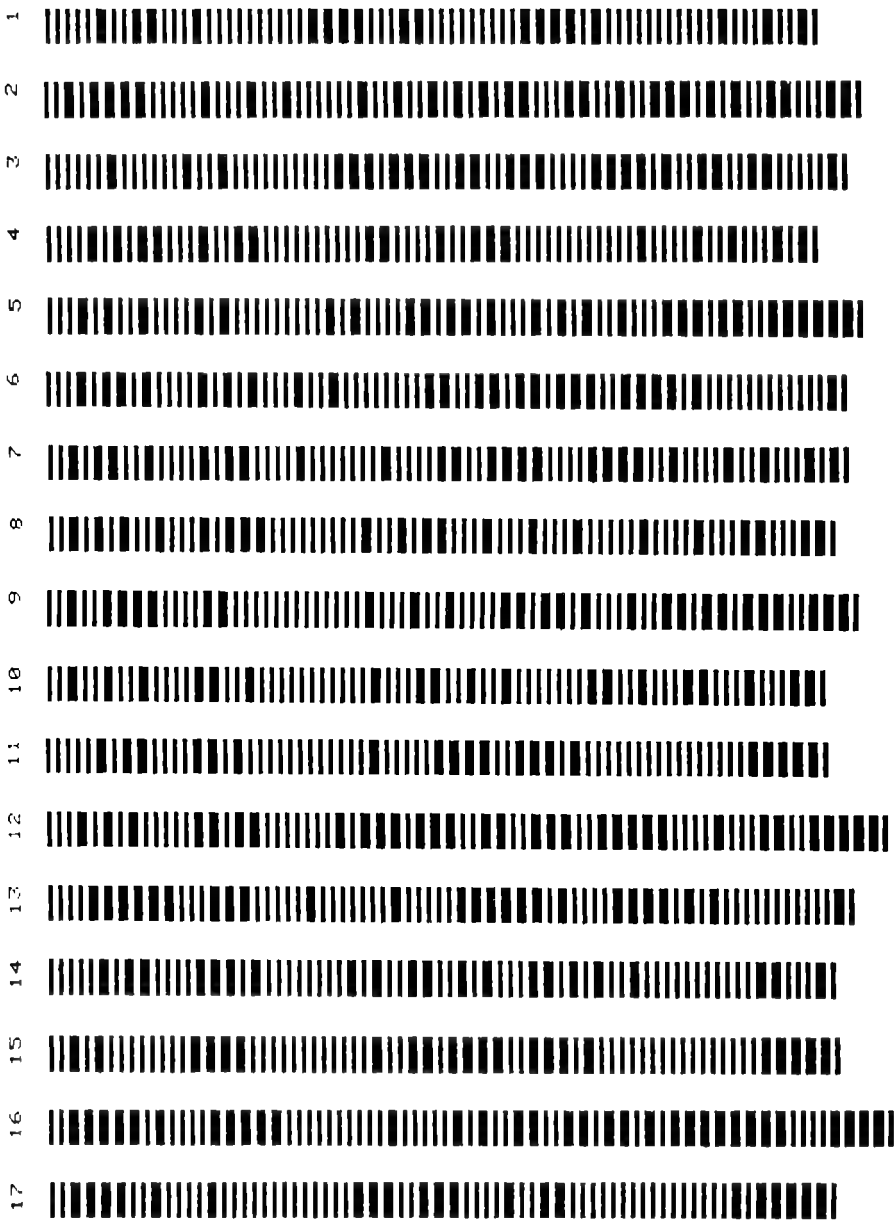
MIS
159 BYTES
= 32 ROWS

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	

5 Division auf unendlich

DIV
85 BYTES
= 17 ROWS



6 Terminverwaltung

TERMIN
496 BYTES
= 100 ROWS

"Die Programme "TERMIN" und "TERMIN?" getrennt hintereinander einlesen und das eventuell in der Mitte stehende "END" löschen."

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

84

36

15

38

39

40 41 42 43 46 49 

4

4 4 

4









7

.....

.....

.....

55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	
70	
71	
72	
73	

93	
94	
95	
96	
97	
98	
99	
100	

TERMIN?
394 BYTES
= 79 ROWS

Figure 1 displays a 17x17 grid of small plots, each representing the spatial distribution of a specific species of the genus *Drosophila*. The rows and columns are labeled with numbers 1 through 17, corresponding to the species. Each plot shows a unique pattern of black dots, indicating the geographic distribution of that species across the study area. The distributions vary significantly between species, with some showing more widespread or clustered patterns than others.

Year	Number of cases
18	100
19	100
20	100
21	100
22	100
23	100
24	100
25	100
26	100
27	100
28	100
29	100
30	100
31	100
32	100
33	100
34	100
35	100
36	100

56 57 58 

59

09

61

62 63 64 

65

69

67 6B 

69






702

71

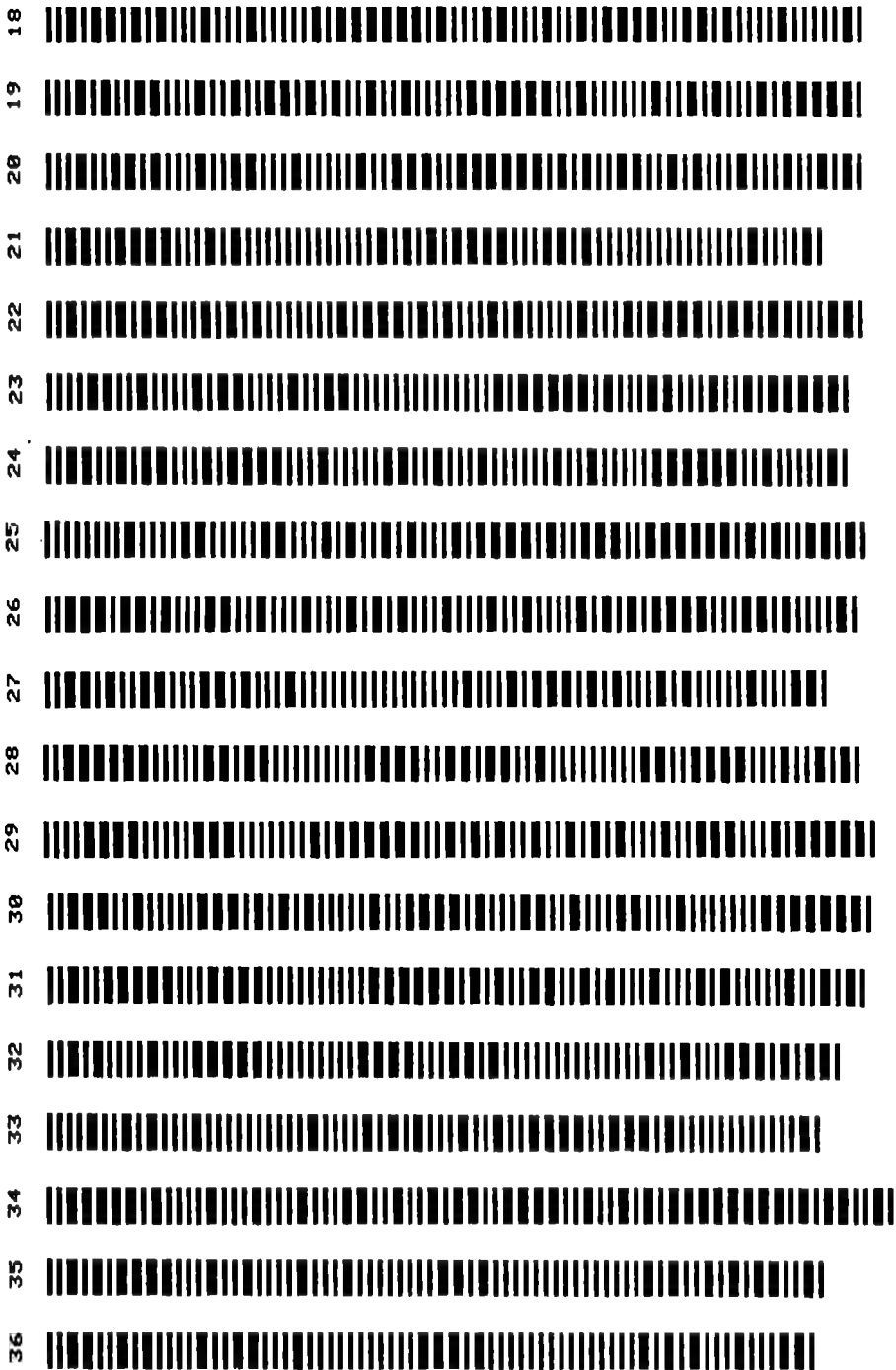
722

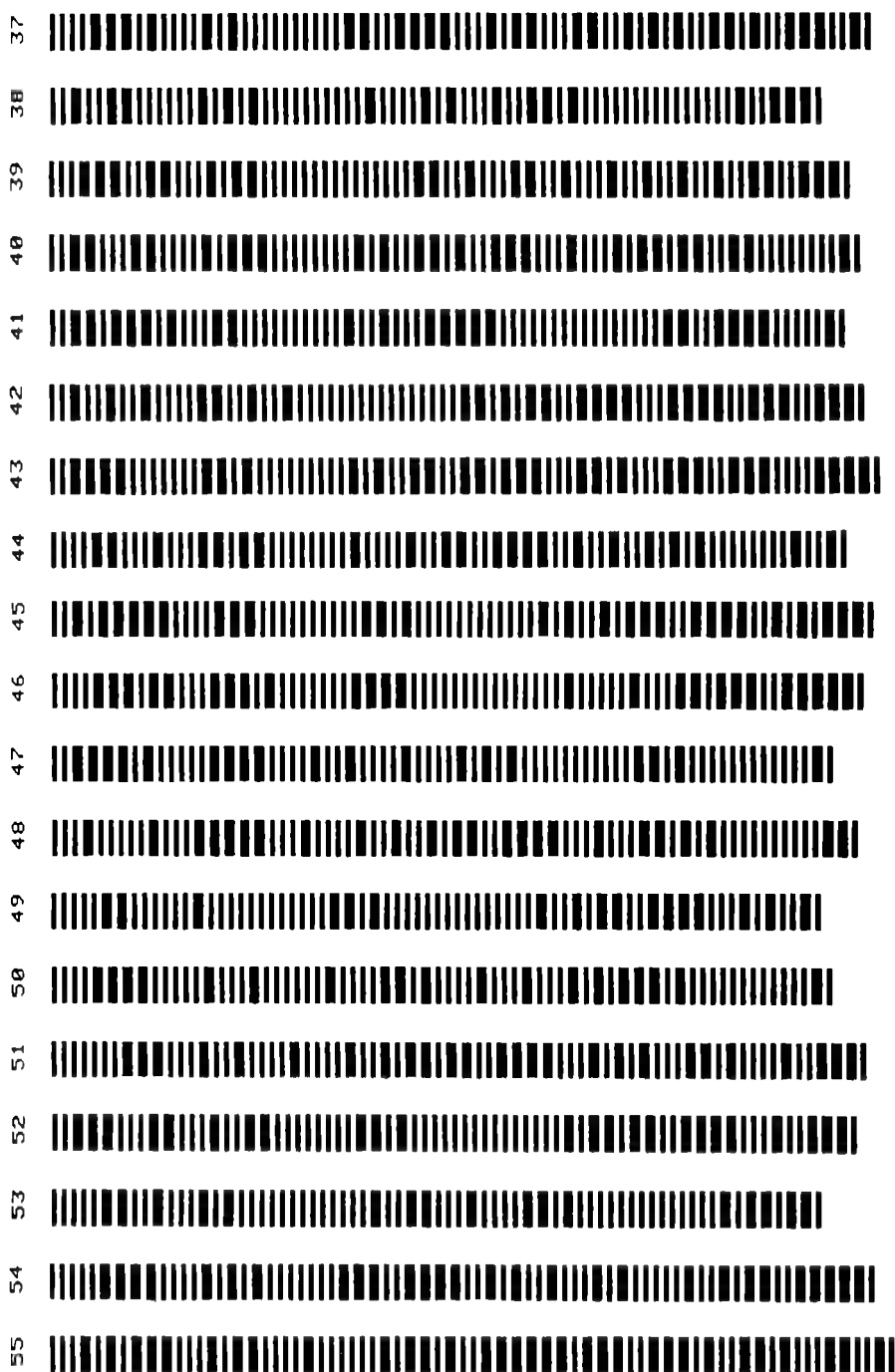
73

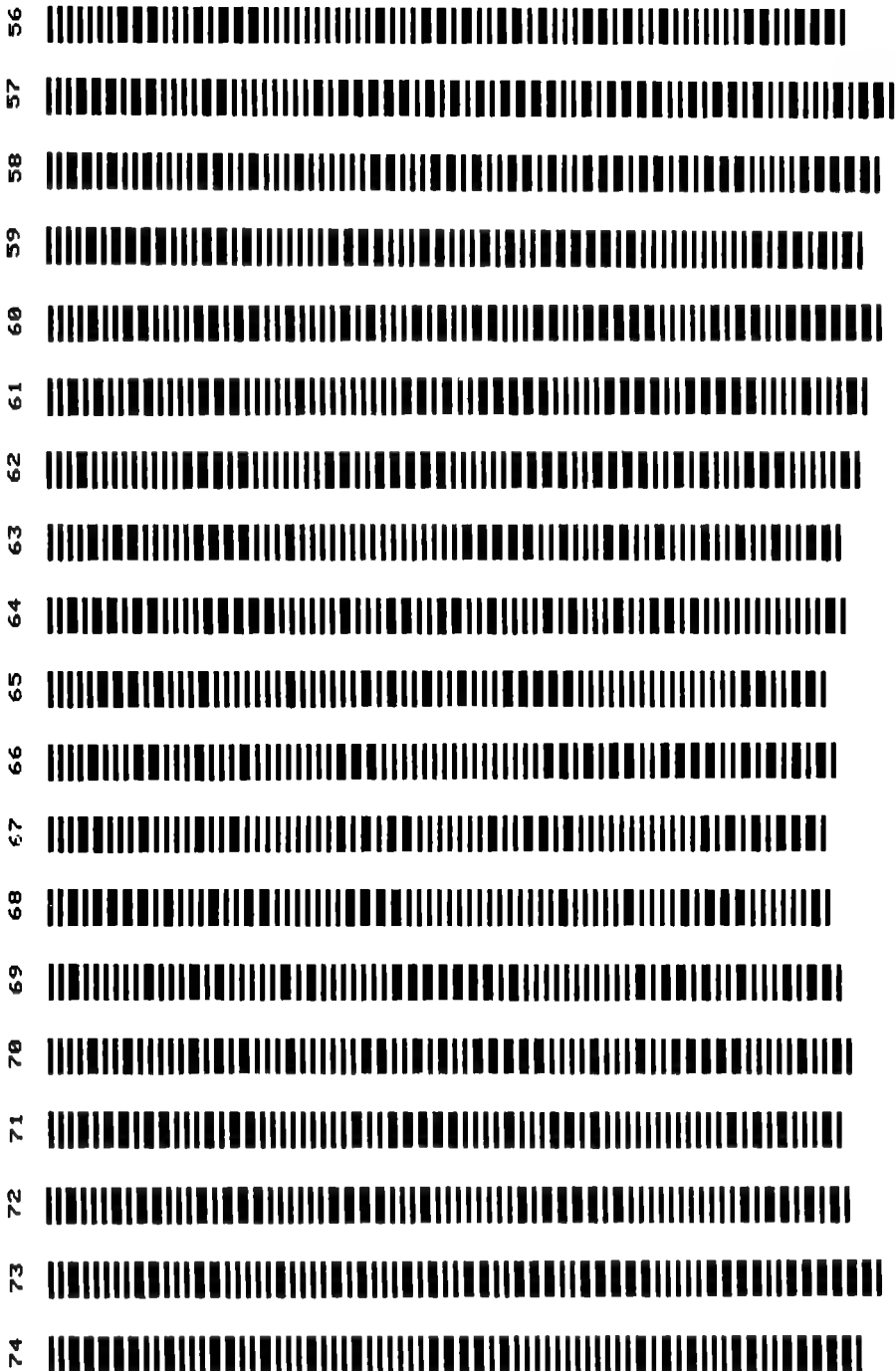
72

75	
76	
77	
78	
79	

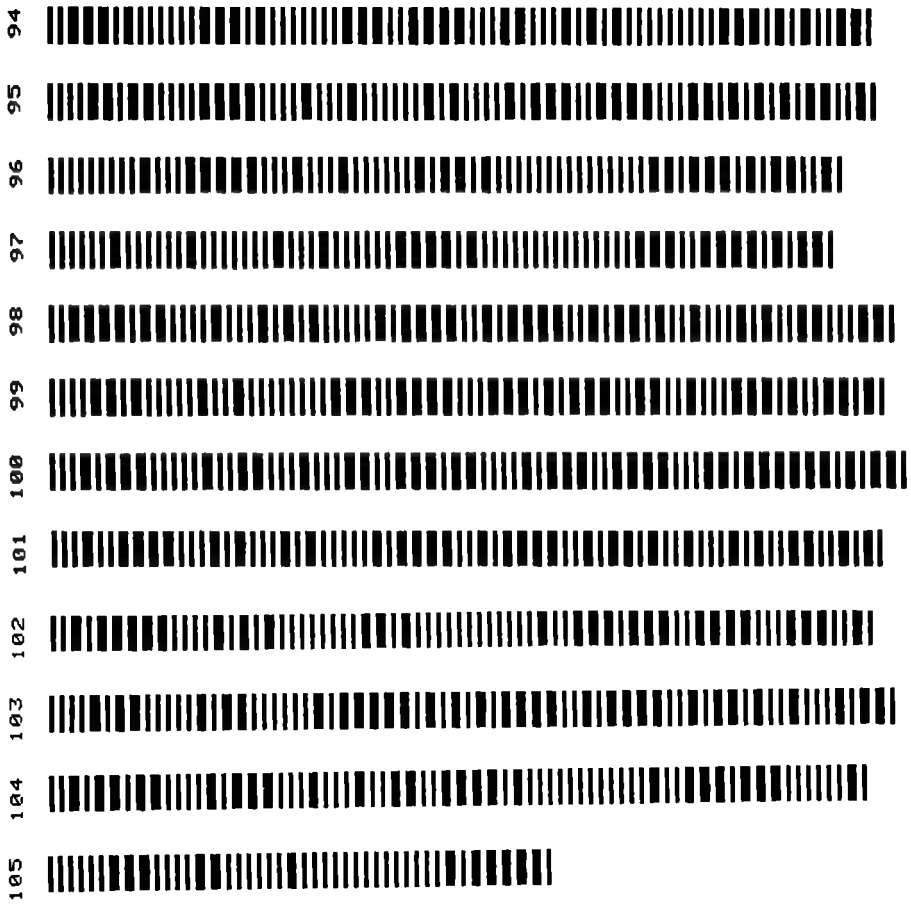
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	







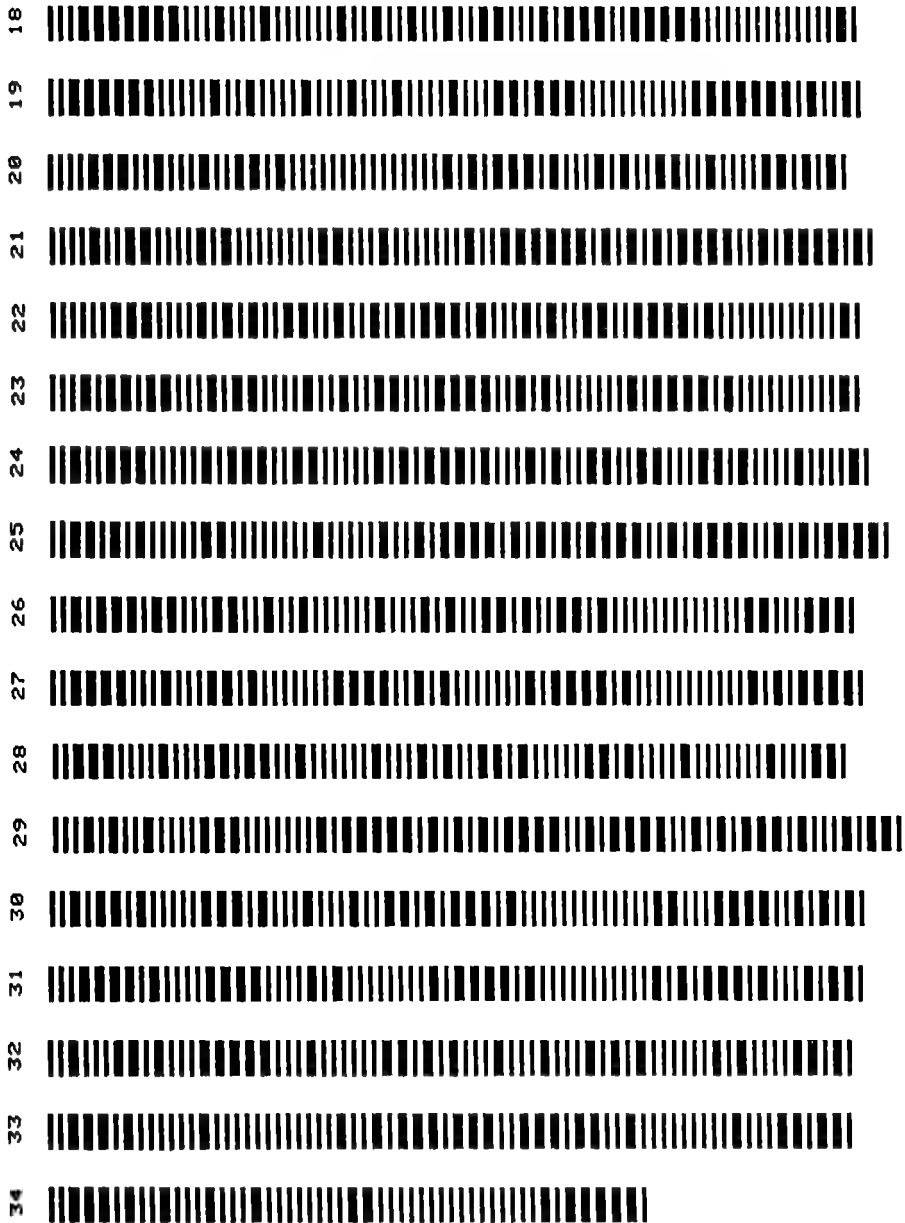
75	
76	
77	
78	
79	
80	
81	
82	
83	
84	
85	
86	
87	
88	
89	
90	
91	
92	
93	



8 Wecken durch den HP-41CX

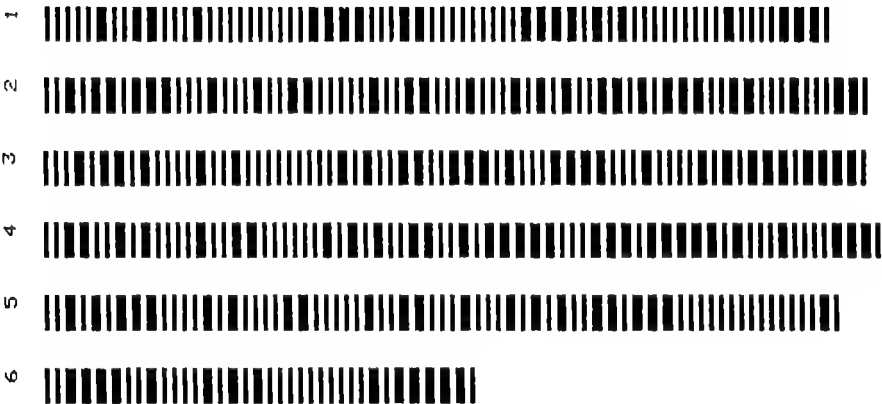
WECKER
168 BYTES
= 34 ROWS

Line	Text
1	1. The first part of the document is a title page.
2	2. The second part of the document is a table of contents.
3	3. The third part of the document is a list of references.
4	4. The fourth part of the document is a list of figures.
5	5. The fifth part of the document is a list of tables.
6	6. The sixth part of the document is a list of appendices.
7	7. The seventh part of the document is a list of footnotes.
8	8. The eighth part of the document is a list of endnotes.
9	9. The ninth part of the document is a list of glossary.
10	10. The tenth part of the document is a list of index.
11	11. The eleventh part of the document is a list of bibliography.
12	12. The twelfth part of the document is a list of references.
13	13. The thirteenth part of the document is a list of figures.
14	14. The fourteenth part of the document is a list of tables.
15	15. The fifteenth part of the document is a list of appendices.
16	16. The sixteenth part of the document is a list of footnotes.
17	17. The seventeenth part of the document is a list of endnotes.



9 Löschen des erweiterten Speichers

CLEN
26 BYTES
= 6 ROWS



CLEMX
41 BYTES
= 9 ROWS

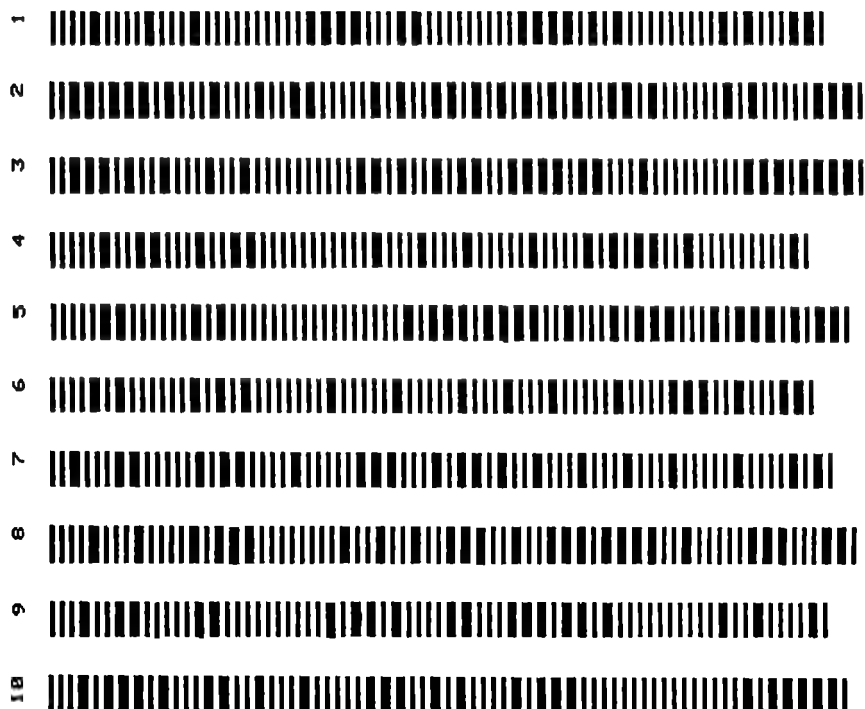
10 Primfaktorzerlegung

PRIM
77 BYTES
= 16 ROWS

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16

11 Quersummenberechnung




















ESUM
50 BYTES
= 16 ROWS








12 Fakultäten, Variationen und Kombinationen
auch für Mengen größer 69

FAX
204 BYTES
= 41 ROWS

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	



















18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	



















37	
38	
39	
40	
41	

13 Bruchrechnen

BRE1
345 BYTES
= 69 ROWS

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	

17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	

35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	
48	
49	
50	
51	
52	

53	
54	
55	
56	
57	
58	
59	
60	
61	
62	
63	
64	
65	
66	
67	
68	
69	

14 Brüche, schnell gekürzt

BRUCH
104 BYTES
= 21 ROWS

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

18 

19 

20 

21 

Vieweg Programmbibliothek Mikrocomputer

Herausgegeben von Harald Schumny

Band 1

**Graphik-Programme für TRS-80
und HP 9830**

Band 2

**Iterationen, Näherungsverfahren,
Sortiermethoden**

BASIC-Programme für
CMB 3032, HP-9830, TRS 80,
Olivetti 6060

Band 3

BASIC und Pascal im Vergleich

Band 4

BASIC-Anwenderprogramme

Band 5

**BASIC-Programme für den
PC-1211/1212**

Band 6

**Programme für den Einplatten-
computer TM 990/189**
(16-Bit-Prozessor TMS 9980 A)

Band 7

PC-1500-Sammlung I

Band 8

Programme für den PC-1251

Band 9

PC-1500-Sammlung II

Band 10

PC-1500-Sammlung III

Band 11

**Anwenderprogramme zum ZX-81
und ZX-Spectrum**

Band 12

17 Spiele für den PC-1500 A

Band 13

Ausgewählte BASIC-Computerspiele
(Atari 800)

Band 14

Lineares Optimieren
11 HP 41 Programme

Band 15

Dienstprogramme (Tool-Kit)
für den HP-41

Band 16

Geodätische Berechnungsmethoden
(Standard-BASIC)

Vieweg Programmbibliothek Mikrocomputer

Herausgegeben von Harald Schumny

Band 17

**Gelenkgetriebe für die
Handhabungs- und Robotertechnik**
(HP-41 CV)

Band 18

Probleme der Festigkeitslehre
(HP-41-Programme)

Band 19

PC-1500-Sammlung IV

Band 20

**Dienstprogramme (Tool-Kit)
für CBM 4032/8032**

Band 21

HP-41 in der Praxis

Band 22

PC-1500-Sammlung V

Band 23

HP-41-Sammlung

Band 24

PC-1500-Sammlung VI

Band 25

Soziogramm mit dem Commodore 64

Band 26

Mathematikpaket für PC-1500 A

Band 27

Kryptologie-Programme
(HP-41 C/CV)

Band 28

HP-41 – Hilfen und Anwendungen

Band 29

**BASIC-Programme zur Regelungs-
technik** (für den Sirius)

Band 30

Börse
Aktienanalyse mit dem
Commodore 64

Band 31

PC-1500-Sammlung VII

Band 32

**Programme aus Biologie, Chemie,
Geographie mit dem Sharp PC-1211**

Band 33

**Mathematikprogramme für den
Atari 800 XL**

Band 34

PC-1500-Sammlung VIII

Band 35

Schachtrainer PC-1500 A

INFO-KARTE

Bitte informieren Sie mich über die neue Vieweg-Infoware

Ich habe einen Mikrocomputer:

Typ: _____ Hersteller: _____

Ich benutze den Mikrocomputer

hauptsächlich privat

☐

hauptsächlich beruflich

☐

Vorwiegend wende ich den Mikrocomputer in folgenden Bereichen an:

Diese Karte entnahm ich dem Buch:

Schumny (Hrsg.), HP-41. Hilfen und Anwendungen

Meine Buchhandlung:

Gleichzeitig bestelle ich:

Ex. _____

Ex. _____

Anschrift

Name

Vorname

Beruf/Funktion

Straße

PLZ

Ort

VIEWEG-INFOWARE

Lieber Leser,

Sie haben ein Mikrocomputer-Buch aus dem Vieweg-Infoware-Angebot gekauft. Wir wünschen Ihnen viel Freude beim Lesen und Erfolg beim Durcharbeiten.

Gern informieren wir Sie in Zukunft über Neuerscheinungen. Wenn Sie daran interessiert sind, schicken Sie uns bitte diese Karte ausgefüllt zurück.

Wenn Sie an der Weiterentwicklung der Vieweg-Infoware mitarbeiten wollen, z. B. durch Veröffentlichung ausgetesteter Programme zu bestimmten Anwendungsgebieten, dann schreiben Sie uns.

Mit freundlichen Grüßen
Verlag Vieweg

Bitte
mit
60 Pt.
freimachen

Antwortkarte

Friedr. Vieweg & Sohn
Verlagsgesellschaft mbH

Postfach 5829

D - 6200 Wiesbaden 1

VIEWEG PROGRAMMBIBLIOTHEK

Mikrocomputer

Die Bände der Programmbibliothek enthalten ausgetestete Programme zu jeweils einem ausgewählten Themenschwerpunkt oder für einen aktuellen Mikrocomputer. Dabei wird der jeweilige Entwicklungsstand der Rechnertechnik berücksichtigt.

Die Programme sind, ausgehend von einer konkreten Aufgabenstellung, in der Regel in ihrem Ablauf beschrieben und durch ausgeführte Beispiele ergänzt. Wenn es nötig scheint, sind auch theoretische Grundlagen für die Programmierung erläutert.

Durch die graphischen, tabellarischen oder in Textform gegebenen Ablaufbeschreibungen wird die Übertragbarkeit auf andere Rechnertypen erleichtert, so daß die wirtschaftliche Nutzung der einzelnen Bände möglich ist. An Hand gleichartiger Aufgabenstellungen wird fallweise auch die unterschiedliche Arbeitsweise verschiedener Rechnertypen aufgezeigt.

Der Herausgeber bemüht sich ständig um eine sorgfältige Auswahl und Begutachtung der eingesandten Programme. Trotzdem kann keine Gewährleistung für vollständige Fehlerfreiheit übernommen werden. Programme zeigen ja oft erst nach vielen Testläufen mit wechselnden Parametern und Grenzbedingungen logische Fehlreaktionen und Sackgassen.

Für die Fälle, die zu Anregungen oder Kritik führen, sind in jedem Band die Anschriften der einzelnen Autoren angegeben. Wir erhoffen uns dadurch einen regen Gedankenaustausch zwischen Autoren und Benutzern der Programmbibliothek, der sich für beide Seiten als nützlich erweisen dürfte.